**AGENDA**

219th ENGINEERS CANADA BOARD MEETING

April 5, 2023 | 11:00am – 1:00pm ET

Virtual delivery | Zoom details are provided via outlook calendar invitation

Reference materials: Board Policy Manual | Bylaw | Corporate Risk Profile | Strategic Plan

| | | |
|---|---|---|
| **1.** | **Opening** | |
| | **1.1 Call to order and approval of agenda** – K. Baig (pages 1 – 2)<br>*THAT the agenda be approved and the President be authorized to modify the order of discussion.* | |
| | **1.2 Declaration of conflict of interest** (pages 3 - 4) | |
| **2.** | **Board business/required decisions** | |
| | **2.1 2022 audited financial statements** – A. Arenja (pages 5 - 24)<br>*THAT the Board, on recommendation of the FAR Committee, approve the Engineers Canada financial statements for the year ending December 31, 2022, as audited by KMPG LLP, and be placed before the Members at the 2023 Annual Meeting of Members.* | |
| **3.** | **CEQB products** (pages 25 – 48) | |
| | **3.1 Engineers Canada paper on professional practice in software engineering**<br>*THAT the Board, on the recommendation of the CEQB, approve the revised Engineers Canada Paper on professional practice in software engineering (public distribution).* | |
| **4.** | **Next meetings** | |
| | **Board meetings** | |
| | • May 26, 2023 (Halifax, NS)<br>• June 19, 2023 (Niagara-on-the-Lake, ON)<br>• October 5, 2023 (Ottawa, ON) | • December 4, 2023 (virtual)<br>• March 1, 2024 (Ottawa, ON)<br>• April 3, 2024 (virtual) |
| | **Committee meetings** | |
| | • FAR Committee: May 11, 2023 (virtual)<br>• Strategic Planning Task Force: May 16, 2023 (virtual) | • HR Committee (2023-2024): May 27, 2023 (Halifax, NS)<br>• All 2023-2024 committees and task forces: June 19, 2023 (Ontario) |
| **5.** | **In-camera sessions** | |
| | **5.1 Board Directors and CEO**<br>*THAT the meeting move in-camera and be closed to the public at the recommendation of the Board. The attendees at the in-camera session shall include Board Directors and the Engineers Canada CEO.* | |
| | **5.2 Board Directors only**<br>*THAT the meeting move in-camera and be closed to the public at the recommendation of the Board. The attendees at the in-camera session shall include Board Directors.*<br>• *Meeting evaluation* | |
| **6.** | **Closing** (motion not required if all business has been completed) | |

# Board support document

## Meeting norms

Virtual participation:

- Board members and Direct Reports are asked to "show up" to the meeting a few minutes early to test their audio and video connections and are encouraged to reach out to Boardsupport@engineerscanada.ca in advance if they anticipate any connection or technological issues.
- To increase meeting engagement and participation, Board members and Direct Reports are requested to turn on their cameras during the meeting, when possible. All participants will have control over their ability to mute their line upon joining the meeting. Participants are asked to self-mute when they are not speaking to minimize background noise. If a participant is muted by an organizer, this is because there was feedback on the line.
- Participants are asked to use the self-mute function and turn off their cameras, instead of leaving the meeting during all breaks. This will help minimize any technical issues and disruption upon re-connection.
- The "Raise hand" function is only to be used if a participant wishes to ask questions and/or make comments after presentations or during debate. Depending on the Zoom version, participants may find the 'Raise hand' button under "Reactions" or "Participants". Participants should reach out in "Chat" if they are not able to locate it.
- If a participant wishes to speak and have not been called upon or are unable to use the "Raise hand" function, they should say their name with an un-muted microphone and obtain permission from the Chair before speaking.
- The "Chat" function will only be monitored by the offsite AV personnel in respect of technical difficulties. Non-technical questions asked through the "Chat" function will not be answered during the meeting.

To conduct the meeting with reasonable time and fairness:

1. For all motions, the meeting chair will call for abstentions and negative votes from the Directors. Directors who do not state a negative vote or an abstention will be considered in favour of the motion. If, for whatever reason, Directors are unable to speak during the motion and feel their opinion was not heard, they should raise their hand, or reach out in "Chat" for technical support.
2. Wordsmithing of motion texts should be avoided as much as possible so that the meeting can stay on track. If the proposed motion and related decision is understood, the Board should move to a debate and discussion on the proposal and should not focus attention on perfecting the text.
3. Participants are asked to speak for a maximum of two (2) minutes at a time (a timer will be projected on the screen) and will be limited to two (2) chances to speak on any one issue or motion. An opportunity to speak a second time will be granted only after everyone has had a chance to speak. The meeting chair reserves the right to allow additional chances to speak, as necessary.
4. Restating or reiterating the same point is strongly discouraged.
5. In the virtual environment where meeting participants are not able to demonstrate their agreement by nodding, they are encouraged to use the "Reaction" buttons to identify their informal support of others' statements. A safe and respectful environment is encouraged at all times.
6. At the opening of the meeting, the meeting chair will announce which individual will be monitoring the show of hands. The chair will try to ensure that anyone with a raised hand has their point addressed.

# Board support document

## Conflicts of interest

Board members and members of Board committees have an ongoing obligation to identify and disclose actual, reasonably perceived, and potential conflicts of interest. These obligations are set out in case law and are also codified in statute, under the *Canada Not-for-profit Corporations Act* ("CNCA").

While not expressly defined in the CNCA, a conflict of interest is understood to comprise any situation where:

a) an individual's personal interests, or
b) those of a close friend, family member, business associate, corporation, or partnership in which the individual holds a significant interest, or a person to whom the individual owes an obligation, could influence their decisions and impair their ability to:
   i. act in the best interests of the corporation, or
   ii. represent the corporation fairly, impartially, and without bias.

Conflicts of interest exist if a Director's decision could be, or could appear to be, influenced. *It is not necessary that influence actually takes place.* In cases where Directors are in an actual, perceived, or potential conflict of interest, they are required to disclose the conflicting interest to the Board[1] or, in the case where membership approval is sought, to the members,[2] as well as abstain from voting.

## Handling conflicts of interest

Directors may use the following checklist when faced with a situation in which they think they might have an actual, perceived, or potential conflict of interest.

**Step 1 - Identify the matter or issue being considered and the potential conflicting situation in which you are involved.**

E.g. There is an item before the Board requiring discussion and a decision that involves potential litigation between Engineers Canada and the Engineering Regulator with whom you are licensed. Whether or not you are in a conflict of interest is not automatic—it will depend upon the personal circumstances of each Director.

**Step 2 – Assess whether a conflict of interest exists or may exist.**

In assessing whether you have an actual, reasonably perceived or potential conflict of interest, it may be helpful to ask yourself the following questions:

- ☐ Would I, or anyone associated with me benefit from, or be detrimentally affected by my proposed decision or action?
- ☐ Could there be benefits for me in the future that could cast doubt on my objectivity?
- ☐ Do I have a current or previous personal, professional, or financial relationship or association of any significance with an interested party?

[1] Section 141(1) and (2) of the CNCA
[2] Section 141(9)(a) of the CNCA

- Would my reputation or that of a relative, friend, or associate stand to be enhanced or damaged because of the proposed decision or action?
- Do I or a relative, friend, or associate stand to gain or lose financially in some way?
- Do I hold any personal or professional views or biases that may lead others to reasonably conclude that I am not an appropriate person to deal with the matter?
- Have I made any promises or commitments in relation to the matter?
- Have I received a benefit or hospitality from someone who stands to gain or lose from my proposed decision or action?
- Am I a member of an association, club, or professional organization, or do I have particular ties and affiliations with organizations or individuals who stand to gain or lose by my proposed decision or action?
- Could this situation have an influence on any future employment opportunities outside my current duties?
- Could there be any other benefits or factors that could cast doubts on my objectivity?
- Am I confident of my ability to act impartially in the best interests of Engineers Canada?

What perceptions could others have?

- What assessment would a fair-minded member of the public make of the circumstances?
- Could my involvement on this matter cast doubt on my integrity or on Engineers Canada's integrity?
- If I saw someone else doing this, would I suspect that they have a conflict of interest?
- If I did participate in this action or decision, would I be happy if my colleagues and the public became aware of my involvement?
- How would I feel if my actions were highlighted in the media?

**Step 3 – Is the duty to disclose triggered?**

If, in assessing the situation, you determine that you are in an actual, potential, or reasonably perceived conflict of interest, your duty to disclose is triggered. Directors disclosing a conflict must make the disclosure at the meeting at which the proposed contract or transaction is first considered and should request to have the disclosure entered into the minutes of the meeting.[3]

Disclosure must be made of the nature and extent of the interest that you have in the contract or transaction (or proposed contract or transaction).[4] The limited case law dealing with the nature and scope of the disclosure required by a conflicted Director suggests that disclosure must make the other Directors fully informed of the real state of affairs (e.g. what your interest is and the extent of the interest).[5] It will rarely suffice to simply declare that you have a conflict of interest.

**Step 4 – What next?**

Subject to limited exceptions, the general rule is that a conflicted Director cannot vote on the approval of a proposed contract or transaction, even where their interest is adequately disclosed.[6] Further, as a best practice, they should leave the room and not participate in the salient part of the Board meeting.

3 Section 141(1) of the CNCA
4 Section 141(1) and 141(9)(b) of the CNCA
5 *Gray v. New Augarita Porcupine Mines Ltd.,* 1952 CarswellOnt 412 (Jud. Com. of Privy Coun.)
6 Section 141(5) of the CNCA

**BRIEFING NOTE:** For decision

| 2022 audited financial statements | 2.1 |
|---|---|
| Purpose: | To approve the 2022 audited financial statements |
| Link to the Strategic Plan / Purposes: | Board responsibility: Ensure the CEO maintains and acts on a robust and effective risk management system which reflects the Board's risk tolerance level and directs Board approved mitigation strategies |
| Link to the Corporate Risk Profile: | Financial compliance (operational risk) |
| Motion(s) to consider: | *THAT the Board, on recommendation of the FAR Committee, approve the Engineers Canada financial statements for the year ending December 31, 2022, as audited by KMPG LLP, and be placed before the Members at the 2023 Annual Meeting of Members.* |
| Vote required to pass: | Simple majority |
| Transparency: | Open session |
| Prepared by: | D. Menard, Director, Finance |
| Presented by: | A. Arenja, Director from Ontario, and Chair of the FAR Committee |

## Problem/issue definition

- The *Canada Not-for-profit Corporations Act* (CNCA) requires that the corporation's financial statements be placed before the Members at every annual meeting.
- The 2022 audit was performed in February 2023, after the close of year-end.
- KPMG offered the opinion that the "financial statements present fairly, in all material aspects, the financial position of Engineers Canada as at December 31, 2022, and its results of operations, changes in net assets and its cash flows for the year then ended in accordance with Canadian Accounting standards for not-for-profit organizations".
- KPMG's audit findings report found no issues of going concern, no corrected / uncorrected audit misstatements, or control deficiencies.

## Proposed action/recommendation

- The Finance, Audit, and Risk (FAR) Committee proposes that the Board approve the audited financial statements. Thereafter, they shall be presented to the Members at the 2023 Annual Meeting of the Members for information.

## Other options considered

- None. To comply with the CNCA requirements, the Members must receive the financial statements not less than 21 days and not more than 60 days before the annual meeting is held.

## Risks

- Failure to approve the audited financial statements and place them before the Members would be a breach of the CNCA.

## Financial implications

- None.

## Benefits

- Members will remain informed on the financial position of the organization, and Engineers Canada remains in compliance with CNCA requirements.

## Consultation

- The FAR Committee met on December 14, 2022 with KPMG LLP, the public accountants (re-appointed by the Members in 2022), to discuss the proposed audit plan.
- The FAR Committee met with the KPMG auditors on March 10, 2023 to review the draft financial statements and the audit findings report.

## Next steps (if motion approved)

- Draft 2022 audited financial statements to be circulated to the Members with the AMM agenda book.

## Appendix

- **Appendix 1:** 2022 draft audited financial statements

Financial Statements of

# ENGINEERS CANADA

And Independent Auditor's Report thereon

Year ended December 31, 2022

# INDEPENDENT AUDITOR’S REPORT

To the Members of Engineers Canada

## *Opinion*

We have audited the financial statements of Engineers Canada, which comprise:

- the statement of financial position as at end of December 31, 2022
- the statement of operations for the year then ended
- the statement of changes in net assets for the year then ended
- the statement of cash flows for the year then ended
- and notes to the financial statements, including a summary of significant accounting policies

(Hereinafter referred to as the "financial statements").

In our opinion, the accompanying financial statements, present fairly, in all material respects, the financial position of Engineers Canada as at December 31, 2022, and its results of operations, its changes in net assets and its cash flows for the year then ended in accordance with Canadian accounting standards for not-for-profit organizations.

## *Basis for Opinion*

We conducted our audit in accordance with Canadian generally accepted auditing standards. Our responsibilities under those standards are further described in the ***"Auditor’s Responsibilities for the Audit of the Financial Statements"*** section of our auditor’s report.

We are independent of Engineers Canada in accordance with the ethical requirements that are relevant to our audit of the financial statements in Canada and we have fulfilled our other responsibilities in accordance with these ethical requirements.

We believe that the audit evidence we have obtained is sufficient and appropriate to provide a basis for our opinion.

### Responsibilities of Management and Those Charged with Governance for the Financial Statements

Management is responsible for the preparation and fair presentation of the financial statements in accordance with Canadian accounting standards for not-for-profit organizations, and for such internal control as management determines is necessary to enable the preparation of financial statements that are free from material misstatement, whether due to fraud or error.

In preparing the financial statements, management is responsible for assessing Engineers Canada's ability to continue as a going concern, disclosing as applicable, matters related to going concern and using the going concern basis of accounting unless management either intends to liquidate Engineers Canada or to cease operations, or has no realistic alternative but to do so.

Those charged with governance are responsible for overseeing Engineers Canada's financial reporting process.

### Auditor’s Responsibilities for the Audit of the Financial Statements

Our objectives are to obtain reasonable assurance about whether the financial statements as a whole are free from material misstatement, whether due to fraud or error, and to issue an auditor’s report that includes our opinion.

Reasonable assurance is a high level of assurance, but is not a guarantee that an audit conducted in accordance with Canadian generally accepted auditing standards will always detect a material misstatement when it exists.

Misstatements can arise from fraud or error and are considered material if, individually or in the aggregate, they could reasonably be expected to influence the economic decisions of users taken on the basis of the financial statements.

As part of an audit in accordance with Canadian generally accepted auditing standards, we exercise professional judgment and maintain professional skepticism throughout the audit.

We also:

- Identify and assess the risks of material misstatement of the financial statements, whether due to fraud or error, design and perform audit procedures responsive to those risks, and obtain audit evidence that is sufficient and appropriate to provide a basis for our opinion.

  The risk of not detecting a material misstatement resulting from fraud is higher than for one resulting from error, as fraud may involve collusion, forgery, intentional omissions, misrepresentations, or the override of internal control.

- Obtain an understanding of internal control relevant to the audit in order to design audit procedures that are appropriate in the circumstances, but not for the purpose of expressing an opinion on the effectiveness of Engineers Canada's internal control.
- Evaluate the appropriateness of accounting policies used and the reasonableness of accounting estimates and related disclosures made by management.
- Conclude on the appropriateness of management's use of the going concern basis of accounting and, based on the audit evidence obtained, whether a material uncertainty exists related to events or conditions that may cast significant doubt on Engineers Canada's ability to continue as a going concern. If we conclude that a material uncertainty exists, we are required to draw attention in our auditor's report to the related disclosures in the financial statements or, if such disclosures are inadequate, to modify our opinion. Our conclusions are based on the audit evidence obtained up to the date of our auditor's report. However, future events or conditions may cause Engineers Canada to cease to continue as a going concern.
- Evaluate the overall presentation, structure and content of the financial statements, including the disclosures, and whether the financial statements represent the underlying transactions and events in a manner that achieves fair presentation.
- Communicate with those charged with governance regarding, among other matters, the planned scope and timing of the audit and significant audit findings, including any significant deficiencies in internal control that we identify during our audit.

Chartered Professional Accountants, Licensed Public Accountants

Ottawa, Canada

(date)

# ENGINEERS CANADA

Statement of Financial Position

December 31, 2022, with comparative information for 2021

| | 2022 | 2021 |
|---|---|---|
| **Assets** | | |
| Current assets: | | |
| Cash (note 3) | $ 2,102,176 | $ 3,037,065 |
| Amounts receivable (note 4) | 1,193,477 | 1,197,114 |
| Prepaid expenses and deposits | 267,510 | 222,338 |
| | 3,563,163 | 4,456,517 |
| Investments (note 5) | 15,760,893 | 16,638,837 |
| Tangible capital assets (note 6) | 641,984 | 662,447 |
| | $ 19,966,040 | $ 21,757,801 |
| **Liabilities and Net Assets** | | |
| Current liabilities: | | |
| Accounts payable and accrued liabilities (note 7) | $ 551,399 | $ 692,117 |
| Deferred contributions | 74,380 | 87,142 |
| | 625,779 | 779,259 |
| Deferred lease inducement (note 8) | 306,086 | 393,539 |
| Net assets (note 9): | | |
| Internally restricted: | | |
| Contingency reserve | 2,500,000 | 2,500,000 |
| Legal contingency reserve | 1,500,000 | 1,500,000 |
| Strategic priorities reserve | 2,000,000 | 2,000,000 |
| Invested in tangible capital assets | 492,588 | 470,366 |
| Unrestricted | 12,541,587 | 14,114,637 |
| | 19,034,175 | 20,585,003 |
| Commitments (note 10) | | |
| | $ 19,966,040 | $ 21,757,801 |

See accompanying notes to financial statements.

On behalf of the Board:

_______________________ Director _______________________ Director

# ENGINEERS CANADA

Statement of Operations

Year ended December 31, 2022, with comparative information for 2021

| | 2022 | 2021 |
|---|---|---|
| Revenue: | | |
| National programs (note 11) | $ 9,968,571 | $ 9,824,255 |
| Corporate services | 3,279,227 | 3,124,386 |
| Investment income | 589,431 | 312,826 |
| Outreach | 22,600 | 17,600 |
| Unrealized gain (loss) in investments | (2,298,681) | 1,179,903 |
| | 11,561,148 | 14,458,970 |
| Expenses: | | |
| Operating expenses: | | |
| Accreditation | 321,241 | 88,391 |
| Fostering working relationships | 148,226 | 1,938 |
| Services and tools | 107,494 | 123,500 |
| National programs | 882,694 | 884,668 |
| Advocating to the Federal government | 65,511 | 44,589 |
| Research and regulatory changes | 2,525 | 20,213 |
| International mobility | 66,816 | 58,216 |
| Promotion and outreach | 449,343 | 186,686 |
| Diversity and inclusion | 167,178 | 208,141 |
| Protect official marks | 156,746 | 132,996 |
| Secretariat services | 1,190,269 | 232,073 |
| Corporate services (note 12) | 7,354,726 | 6,982,816 |
| | 10,912,769 | 8,964,227 |
| Excess of revenue over expenses before the undernoted | 648,379 | 5,494,743 |
| Projects spending: | | |
| Accreditation improvement project | 136,318 | 221,574 |
| International mobility - IIDD one-time project | 2,560 | 54,599 |
| Services and tools – competency-based assessment | 70,239 | 214,592 |
| Service and tools – NMDB | 184,040 | 173,110 |
| Investigate and validate the purpose and scope of accreditation | 992,578 | 12,360 |
| Reinforce trust and the value of licensure | 374,785 | 4,575 |
| Strengthen collaboration and harmonization | 95,459 | – |
| Accelerate 30 by 30 | 208,953 | – |
| Research and regulatory changes –foresight | 133,675 | – |
| Mobility register improvements | 600 | – |
| | 2,199,207 | 680,810 |
| Excess (deficiency) of revenue over expenses | $ (1,550,828) | $ 4,813,933 |

See accompanying notes to financial statements.

# ENGINEERS CANADA

Statement of Changes in Net Assets

Year ended December 31, 2022, with comparative information for 2021

| | Contingency (note 9) | Legal contingency reserve (note 9) | Strategic priorities reserve (note 9) | Invested in tangible capital assets | Unrestricted | 2022 Total | 2021 Total |
|---|---|---|---|---|---|---|---|
| Balance, beginning of year | $ 2,500,000 | $ 1,500,000 | $ 2,000,000 | $ 470,366 | $ 14,114,637 | $ 20,585,003 | $ 15,771,070 |
| Excess (deficiency) of revenue over expenses | – | – | – | – | (1,550,828) | (1,550,828) | 4,813,933 |
| Amortization of tangible capital assets | – | – | – | (180,841) | 180,841 | – | – |
| Additions to tangible capital assets | – | – | – | 160,378 | (160,378) | – | – |
| Amortization of leasehold inducement | – | – | – | 42,685 | (42,685) | – | – |
| Balance, end of year | $ 2,500,000 | $ 1,500,000 | $ 2,000,000 | $ 492,588 | $ 12,541,587 | $ 19,034,175 | $ 20,585,003 |

See accompanying notes to financial statements

# ENGINEERS CANADA

Statement of Cash Flows

Year ended December 31, 2022, with comparative information for 2021

| | 2022 | 2021 |
|---|---|---|
| Cash provided by (used in): | | |
| Operating activities: | | |
| Excess (deficiency) of revenue over expenses | $ (1,550,828) | $ 4,813,933 |
| Items not involving cash: | | |
| Amortization of tangible capital assets | 180,841 | 134,735 |
| Amortization of lease inducement | (87,453) | (87,453) |
| Change in net unrealized loss (gain) on investments | 2,298,681 | (1,179,903) |
| Change in non-cash operating working capital: | | |
| Decrease (increase) in amounts receivable | 3,637 | (41,076) |
| Increase in prepaid expenses and deposits | (45,172) | (100,329) |
| Increase (decrease) in accounts payable and accrued liabilities | (140,718) | 126,558 |
| Decrease in deferred contributions | (12,762) | (32,587) |
| | 646,226 | 3,633,878 |
| Investing activities: | | |
| Net purchases of investments | (1,420,737) | (2,741,231) |
| Additions to tangible capital assets | (160,378) | (152,283) |
| | (1,581,115) | (2,893,514) |
| Increase (decrease) in cash | (934,889) | 740,364 |
| Cash, beginning of year | 3,037,065 | 2,296,701 |
| Cash, end of year | $ 2,102,176 | $ 3,037,065 |

See accompanying notes to financial statements.

**1. Governing statutes and nature of operations:**

Engineers Canada is a national federation of the twelve provincial and territorial associations authorized to license engineers and regulate the practice of the profession across Canada. Engineers Canada exists so that constituent associations have support for an advancing engineering profession and its self-regulation in the public interest at a cost that is justified by the results.

Engineers Canada was originally incorporated without share capital under Part II of the Canada Corporations Act. Effective October 31, 2013, Engineers Canada continued its articles of incorporation from Canada Corporations Act to the Canada Not-for-profit Corporations Act and changed its name to Engineers Canada from the Canadian Council of Professional Engineers. Engineers Canada is a not-for-profit organization and as such is exempt from income tax under Section 149(1)(I) of the Income Tax Act (Canada).

**2. Significant accounting policies:**

These financial statements have been prepared by management in accordance with Canadian accounting standards for not-for-profit organizations in Part III of the CPA Canada Handbook - Accounting and include the following significant accounting policies:

(a) Revenue recognition:

Engineers Canada follows the deferral method of accounting for contributions for not-for-profit organizations.

Engineers Canada's principal sources of revenue are provincial assessment fees from members, and amounts from affinity and insurance programs.

Revenues for provincial assessment and annual per capita fees are recognized when the constituent members have been invoiced and are included in corporate services revenue on the statement of operations. Revenues from affinity programs are recognized when the amount becomes collectible according to the terms of the arrangement. These amounts are included in national program revenues on the statement of operations.

Investment income is recognized based on the number of days the investment was held during the year. Dividends are recognized as of the ex-dividend date. Gains or losses on the disposal of investments are determined using the average cost method. All investment revenues including realized and unrealized gains and losses on investments are recognized in the statement of operations.

Externally funded project revenues, which include government funded project revenues, are recognized using the deferral method of accounting as the related eligible expenses are incurred in accordance with the terms of each contract. Amounts received in excess of eligible expenses are disclosed as a liability.

**2. Significant accounting policies (continued):**

(b) Financial instruments:

Financial instruments are recorded at fair value on initial recognition. Equity instruments that are quoted in an active market are subsequently measured at fair value. All other financial instruments are subsequently recorded at cost or amortized cost, unless management has elected to carry the instruments at fair value. Engineers Canada has elected to carry investments at fair value.

Transaction costs incurred on the acquisition of financial instruments measured subsequently at fair value are expensed as incurred. All other financial instruments are adjusted by transaction costs incurred on acquisition and financing costs, which are amortized using straight-line rate method.

Financial assets are assessed for impairment on an annual basis at the end of the fiscal year. Where an indicator of impairment is present, Engineers Canada determines if there is a significant adverse change in the expected amount or timing of future cash flows from the financial asset. If there is a significant adverse change in the expected cash flows, the carrying value of the financial asset is reduced to the highest of the present value of the expected cash flows, the amount that could be realized from selling the financial asset or the amount Engineers Canada expects to realize by exercising its right to any collateral. If events and circumstances reverse in a future period, an impairment loss will be reversed to the extent of the improvement, not exceeding the initial impairment charge.

(c) Tangible capital assets:

Tangible capital assets are recorded at cost less accumulated amortization. When a capital asset no longer contributes to Engineers Canada's ability to provide services, its carrying amount is written down to its residual value.

Amortization of tangible capital assets is provided on the straight-line basis as follows:

| Asset | Terms |
|---|---|
| Furniture, fixtures and equipment | 4 years |
| Computer hardware | 4 years |
| Leasehold improvements | Remaining term of lease |

(d) Deferred lease inducement:

Leasehold inducements are deferred and amortized over the term of the lease. Annual amortization is recorded as a credit to corporate services expense.

**2. Significant accounting policies (continued):**

(e) Allocated expenses:

In the statement of operations, Engineers Canada presents it expenses by function.

Engineers Canada does not allocate expenses between functions subsequent to initial recognition.

(f) Foreign currency translation:

Foreign currency transactions are initially recorded at the rate of exchange prevailing at the date of translation. Thereafter, monetary assets and liabilities are translated at the exchange rate in effect at the statement of financial position date. Revenue and expenses in a foreign currency are translated at the average monthly rate in effect during the year. Gains and losses resulting from the translation are included in investment income in the statement of operations.

(g) Use of estimates:

The preparation of the financial statements requires management to make estimates and assumptions that affect the reported amounts of assets and liabilities and disclosure of contingent assets and liabilities at the date of the financial statements and the reported amounts of revenue and expenses during the year. Actual results could differ from these estimates. These estimates are reviewed annually and as adjustments become necessary, they are recognized in the financial statements in the period they become known.

## 3. Cash:

(a) Cash balances:

Engineers Canada's cash balances consist of operating cash held in Canadian chartered banks and amounts held in a Canadian money market fund, and can be liquidated at any time.

| | 2022 | 2021 |
|---|---|---|
| Operating cash | $ 1,037,730 | $ 1,104,945 |
| Canadian money market fund | 1,064,446 | 2,202,120 |
| | $ 2,102,176 | $ 3,307,065 |

(b) Line of credit

Engineers Canada has a line of credit allowing it to borrow up to $500,000 (2021 - $500,000) at an interest rate of prime plus 1%. This line of credit is subject to annual renewal. There was no outstanding balance as at December 31, 2022 or 2021.

## 4. Amounts receivable:

| | 2022 | 2021 |
|---|---|---|
| Affinity and insurance programs | $ 1,133,900 | $ 1,134,700 |
| Government remittances receivable | 59,577 | 62,130 |
| Due from members | – | 284 |
| | $ 1,193,477 | $ 1,197,114 |

**5. Investments:**

| | 2022 Fair value | 2022 Cost | 2021 Fair value | 2021 Cost |
|---|---|---|---|---|
| Bond funds | $ 6,482,795 | $ 7,002,609 | $ 7,232,321 | $ 7,388,184 |
| Canadian equity funds | 2,554,561 | 2,343,384 | 2,617,659 | 2,153,082 |
| International equity funds | 1,826,600 | 1,780,312 | 2,314,217 | 1,557,593 |
| U.S. equity funds | 2,167,331 | 1,877,256 | 1,884,508 | 1,585,192 |
| Global equity funds | 1,865,534 | 2,053,163 | 2,098,531 | 1,828,163 |
| Alternative – Global | 864,072 | 926,376 | 491,601 | 491,601 |
| | $ 15,760,893 | $ 15,983,100 | $ 16,638,837 | $ 15,003,815 |

Investments are held by Engineers Canada to fund its internally restricted net assets for the purposes specified in Note 9(a).

**6. Tangible capital assets:**

| | Cost | Accumulated amortization | 2022 Net book value | 2021 Net book value |
|---|---|---|---|---|
| Furniture, fixtures and equipment | $ 338,995 | $ 230,612 | $ 108,383 | $ 50,879 |
| Computer hardware | 443,242 | 373,752 | 69,490 | 62,347 |
| Leasehold improvements | 1,186,958 | 722,847 | 464,111 | 549,221 |
| | $ 1,969,195 | $ 1,327,211 | $ 641,984 | $ 662,447 |

Cost and accumulated amortization at December 31, 2021 amounted to $1,808,817 and $1,146,370, respectively.

## 7. Accounts payable and accrued liabilities:

| | | 2022 | | 2021 |
|---|---|---|---|---|
| Operating | $ | 280,161 | $ | 277,915 |
| Accrued liabilities | | 67,754 | | 85,105 |
| Payroll related accruals | | 162,805 | | 288,853 |
| Secondary Professional Liability insurance premiums repayable to members | | 40,679 | | 40,244 |
| | $ | 551,399 | $ | 692,117 |

There are no amounts payable for government remittances such as sales or payroll-related taxes included in operating or accrued liabilities.

## 8. Deferred lease inducement:

In 2015, Engineers Canada entered into a lease agreement to rent premises for the next ten years. As part of this agreement, Engineers Canada received a tenant allowance to cover fit-up costs up to a maximum of $30 per square foot of space rented, as well as a rent-free period for nine months.

| | | Rent-free leasehold inducements | | Tenant allowance - fit-up costs | | Total |
|---|---|---|---|---|---|---|
| Balance, beginning of year | $ | 201,458 | $ | 192,081 | $ | 393,539 |
| Less: amortization | | (44,768) | | (42,685) | | (87,453) |
| Balance, end of year | $ | 156,690 | $ | 149,396 | $ | 306,086 |

## 9. Net assets:

Engineers Canada's overall objective with regard to its net assets is to ensure stability for the delivery of on-going programs and services, to fund strategic initiatives and to mitigate the financial impact of risks to its operations and achievement of strategic objectives. Engineers Canada manages its net assets by establishing restricted funds and committing amounts in the internally restricted net assets for anticipated future strategic priorities, contingencies, legal defense, and other capital requirements. These allocations are presented in the statement of changes in net assets and disclosed in Note 9(a).

**9. Net assets (continued):**

Engineers Canada's objective with respect to unrestricted net assets is to maintain a balance sufficient to meet the needs associated with ongoing operations. Engineers Canada's net assets invested in its capital assets is equal to their net book value less the corresponding lease inducement.

Engineers Canada is not subject to externally imposed capital requirements and it adopted a new overall strategy with respect to net assets that took affect in 2022.

(a) Internally restricted net assets:

Internally restricted net assets are funds committed for specific purposes, which reflect the application of Engineers Canada's Board policy as follows:

The Contingency Reserve is to mitigate the financial impact of the risk of future unexpected, negative events that could have a significant, adverse impact on the operations, revenues, and expenses of Engineers Canada. This reserve has a target level of $2,500,000.

The Legal Reserve is to ensure that funds are available in case of legal challenge, to provide funds to cover deductibles for insurances, and to assist the Engineering Regulators where it is determined that they do not have the financial resources to defend an enforcement action and/or statutory obligation that has a clear and significant impact on the other Regulators. This reserve has a target level of $1,500,000.

The Strategic Priorities Reserve is to provide funds for planned strategic initiatives, and to respond to future risks and investment needs in the performance, accessibility, and security of its information technology assets. This reserve has a target level of $2,000,000.

Engineers Canada's Board of Directors will also create new reserves and/or discontinue existing reserves, if and when required.

**10. Commitments:**

Engineers Canada leases equipment and office space under operating leases which expire in April 2024 and June 2026. The future rental payments over the next four years including operating costs and taxes, are as follows:

| | |
|---|---|
| 2023 | $ 657,968 |
| 2024 | 655,797 |
| 2025 | 655,073 |
| 2026 | 327,287 |
| | $ 2,296,125 |

**11. National programs:**

Engineers Canada is a party to a number of agreements with financial services companies. Under these agreements Engineers Canada derives revenues, referred to in these financial statements as affinity program and secondary professional liability insurance based on the purchase of goods and services by the members of Engineers Canada's various provincial and territorial member associations.

These agreements have varying terms and conditions as well as varying termination dates and methods, some of which have fixed expiry dates with renewal options and some of which are on-going until terminated with notice by either party.

The two most significant agreements account for 92% (2021 - 92%) of the national program revenues and have the following terms:

- twelve-year term expiring December 2029 with automatic five-year renewals until terminated by either party with 180 days' notice prior to the end of any such period which accounts for 76% (2021 - 75%) of the national program revenues; and
- on-going with no fixed expiry date which accounts for 16% (2021 - 16%) of the national program revenues.

**12. Pension plan contributions:**

Engineers Canada is the administrator of the Staff Pension Plan for Employees of Engineers Canada, which is a defined contribution plan registered with Financial Services Commission of Ontario. The contributions to the plan are $374,074 (2021 - $214,494), which are included in corporate services expense.

**13. Comparative information:**

Certain comparative information has been reclassified to conform to the financial statement presentation adopted in the current year.

**14. Financial risk management:**

Engineers Canada is exposed to various financial risks resulting from both operational and investment activities. Engineers Canada's management addresses the situation by having different related policies such as the Reserves Policy, the Financial Commitments and Payment Policy, amongst others. Engineers Canada also outsources the management of its investment portfolio to an outside firm. There have been no significant changes to Engineers Canada's policies, procedures and methods to manage these risks.

(a) Market risk:

Market risk is the risk that the fair value of future cash flows of a financial instrument will fluctuate because of changes in market prices due to currency, interest rate and other price risks. Engineers Canada is exposed to market risk with respect to its investments, as disclosed in Note 5.

(b) Foreign currency risk:

Foreign currency risk is the risk that the fair value or future cash flows of a financial instrument will fluctuate because of changes in foreign exchange rates. Engineers Canada is exposed to foreign currency risk due to its investments denominated in foreign currencies within its US, International and Global equity funds as disclosed in Note 5. Engineers Canada holds minimal cash balances in foreign currencies.

(c) Interest rate risk:

Interest rate risk is the risk that the fair value or future cash flows of a financial instrument will fluctuate due to changes in market interest rates. Engineers Canada is exposed to interest rate risk with respect to its interest-bearing investments as disclosed in notes 3 and 5. Engineers Canada's other financial assets and financial liabilities do not bear significant amounts of interest. Engineers Canada does not use derivative financial instruments to reduce its interest rate risk exposure.

(d) Other price risk:

Other price risk is the risk that the fair value of future cash flows of a financial instrument will fluctuate because of changes in market prices (other than those arising from interest rate risk or foreign currency risk), whether those changes are caused by factors specific to the individual financial instrument or its issuer, or factors affecting all similar financial instruments traded in the market. Engineers Canada is exposed other price risk due to its equity investments as disclosed in Note 5.

(e) Liquidity risk:

Liquidity risk is the risk that Engineers Canada will be unable to fulfill its obligations on a timely or cost-effective manner. Engineers Canada manages its liquidity risk by monitoring its operating requirements. Engineers Canada prepares budget and cash forecasts to ensure it has sufficient funds to fulfill its obligations.

**14. Financial risk management (continued):**

(f) Credit risk:

Credit risk is the risk that one party to a financial instrument will cause a financial loss for the other party by failing to discharge an obligation. Engineers Canada is exposed to credit risk in the event of non-payment by its counterparties in connection with its accounts receivable. In order to mitigate its credit risk, Engineers Canada has entered into long-term agreements for the majority of its receivables, employs credit policies and monitors collection. Refer to Note 11 for further details of the significant counterparty agreements. An allowance for doubtful accounts is established based on factors surrounding the credit risk of specific members, historical trends and other information. At December 31, 2022, the allowance for doubtful accounts was $Nil (2021 - $Nil).

Management believes that Engineers Canada is not exposed to significant risks from its financial instruments, although the interest rate and other price risks have increased in the year due to rising market interest rates and equity market fluctuations.

## BRIEFING NOTE: For decision

| Engineers Canada paper on professional practice in software engineering | 3.1 |
|---|---|
| Purpose: | To approve the revised Engineers Canada paper on professional practice in software engineering |
| Link to the Strategic Plan / Purposes: | Core purpose 6: Actively monitor, research, and advise on changes and advances that impact the Canadian regulatory environment and the engineering profession. |
| Link to Corporate Risk Profile: | Diminished scope and value of engineering regulation (Board risk) |
| Motion(s) to consider: | THAT the Board, on the recommendation of the CEQB, approve the revised Engineers Canada paper on professional practice in software engineering. |
| Vote required to pass: | Simple majority |
| Transparency: | Open session |
| Prepared by: | Ryan Melsom, Manager, Qualifications and CEQB Secretary |
| Presented by: | Margaret Anne Hodges, CEQB Chair |

## Problem/issue definition

- The unregulated practice of software engineering and the unlicensed use of the restricted title "software engineer", "computer engineer", and similar titles that prefix "engineer" within IT-related disciplines, pose significant risks to the protection of the public.
- As with professions such as medicine or law, confusion around title and qualifications misleads the public, as it may cause the public to seek out work from individuals who are not licensed and who do not have the expertise and ethical obligations of the profession. The same is true for engineers.
- The engineering regulators, specifically the National Discipline and Enforcement Officials Group (NDEOG), requested that the CEQB revise *Engineers Canada paper on professional practice in software engineering* (the Paper) to provide a tool in support of the regulation of the practice of software engineering and the use of title in Canada.
- Revisions to this Paper were assigned to the CEQB's Task Force on Software Engineering.
- The revised Paper contributes to two strategic priorities:
    - 1.2 Strengthen collaboration and harmonization.
    - 1.3 Support regulation of emerging areas.

## Proposed action/recommendation

- That the Board, on the recommendation of the CEQB, approve the revised *Engineers Canada paper on professional practice in software engineering*.
- The CEQB provides Engineers Canada papers, among other products, for Regulators with the intent to inform them concisely about a complex issue and present a stance on the matter.
- Papers may be made publicly available or posted on the members-only section of the Engineers Canada website.
- The Paper notes that, excepting a few legislated cases such as aviation engineers, "only engineers (i.e. licensed individuals) can represent themselves as engineers."
- The Paper also defines software engineering practice in relation to the national definition of engineering, provided in the CEQB's 2012 Public guideline on the practice of engineering in Canada:
    - The 'practice of engineering' means any act of planning, designing, composing, evaluating, advising, reporting, directing or supervising, or managing any of the foregoing, that requires the application of

engineering principles and that concerns the safeguarding of life, health, property, economic interests, the public welfare or the environment.

- By highlighting these fundamental linkages between software engineering and other engineering disciplines, the Paper aims to clarify types of software practice that fall under the scope of regulation.

## Risks

- While software engineering practitioners and regulators were consulted during the Paper's development, the Paper has not been validated through broad-based industry consultation.

## Financial implications

- N/A

## Benefits

- The revised Paper takes into consideration the current realities of the software field and uses strong language against the misuse of the software engineer title to protect the public.
- The Paper can be used as a tool for regulators, practitioners, and the public to differentiate software engineering from other non-engineering software work.
- The Paper can help regulators and the public identify when enforcement activities or reporting to engineering regulators should be undertaken against the misuse of the engineering title and unlicensed practice of software engineering.
- This Paper also provides examples of software engineering in four practice areas, discussed in appendices. These examples are not intended to encompass the entire scope and depth of software engineering, but rather provide an interpretation of how software engineering applies in sample areas of practice.
- Appendix E was added to include a link to the letter signed by Engineers Canada and the twelve regulators CEOs, which makes a strong statement on the misuse of the software engineer title.
- APEGA indicated that approval of this Paper is time-sensitive, as it will be of use to them in their regulatory activities.

## Consultation

- In 2019, the CEQB Task Force on Software Engineering was charged with reviewing the Engineers Canada Paper in consultation with the regulators. The task force was comprised of eight members who were either software engineers, academics in software engineering, or individuals regulating the practice of software engineering.
- During a consultation in February 2020, the NDEOG suggested that the topic of cybersecurity and the use of title in engineering profession be included in the Paper, and to consider the 2016 Engineers Canada paper on environmental engineering for content formatting in new appendices.
- In April 2020, it was brought to light that the Paper's content at that time may not be sufficient to address forthcoming significant changes to Quebec's Engineering Act. Once the act was passed into law in September 2020, assessments were made to determine what revisions would be required.
- In response to regulator requests, the task force developed a survey which was delivered to 15 software engineers through an interview format by consultants to help develop practice area-focused appendices. The draft revised Paper and its appendices were approved for consultation by the CEQB in July 2022.
- In addition to feedback from four regulators and the NDEOG through consultation in July 2022, the final Paper was reviewed by a limited number of industry participants from December 2022 to January 2023, which reflected the best process available given the project's resource constraints.
- In alignment with the standard CEQB consultation process, defined under Engineers Canada Board Policy 9.2, most recent revisions to address practitioner feedback have not been reviewed by the officials'

groups and their stakeholders, including the software practitioners that provided that feedback; two members of the task force requested that this be noted as a condition of their approval.

## Next steps (if motion approved)

- The Engineers Canada paper on professional practice in software engineering will undergo legal review, and subsequently be published on the public website.

## Appendices

- **Appendix 1:** Engineers Canada paper on professional practice in software engineering

# Engineers Canada paper on professional practice in software engineering

## 1. INTRODUCTION

In Canada, the profession of engineering is self-regulated by provincial/territorial engineering regulators, pursuant to a statutory mandate set out in engineering legislation. The delegation of regulatory function recognizes the specialized knowledge of the profession and its ability to develop and maintain standards of competency and conduct to ensure that the public interest is served and protected. In fulfilling their statutory mandates, engineering regulators are entrusted with a variety of responsibilities, including regulating the practice of engineering and the use of the engineer title.

***The practice of engineering****:* According to Engineers Canada's "Public guideline on the practice of engineering in Canada", only an engineering licence holder can engage in the independent practice of, or take responsibility for, engineering work, which is defined as any act of planning, designing, composing, evaluating, advising, reporting, directing or supervising, or managing any of the foregoing, that requires the application of engineering principles and that concerns the safeguarding of life, health, property, economic interests, the public welfare, or the environment. This definition serves as a basis for this paper, but it should be noted that the definition of engineering, as well as its enforcement, may vary by regulatory jurisdiction.

***The use of engineer title****:* With a few exceptions, only engineers can represent themselves as engineers. Only engineering licence holders can affix an engineering seal to their work, which demonstrates to the public that an engineer provides the work to the high standards of the engineering profession.

Although each jurisdiction typically establishes its own code of ethics, engineering licence holders are typically bound by their code of ethics to:

- only practice in their areas of competence;
- maintain their knowledge, skills, and abilities throughout their careers; and
- hold paramount the safety, health, and welfare of the public, and the protection of the environment.

To ensure that only competent individuals practice or take responsibility for engineering work, engineering regulators set standards of practice, ethics, and continuing competence. They investigate complaints of unprofessional conduct, and impose disciplinary sanctions including suspension and revocation of an engineering license when appropriate. Engineering regulators also take action against persons who represent themselves as engineers but do not possess an engineering licence or who are actually practising engineering without a licence. The work of engineering regulators protects public interest.

Please consult individual provincial and territorial engineering regulators' acts to learn about the legal requirements applicable in each jurisdiction.

## 2. BACKGROUND

Growing public concerns with automated technology, the increased frequency of malicious cybersecurity events, and the rapid pace with which software is becoming integrated into all aspects of daily life, is drawing increased attention to requirement to hire engineers to protect the public in these areas. This paper defines key elements of the practice of software engineering and explains the legal requirement for this work to be undertaken by engineers. The intended outcome is that regulators, practitioners, and the public are better able to differentiate software engineering from other non-engineering software work. This guidance is provided to help regulators and the public identify when enforcement activities or reporting to engineering regulators should be undertaken against the misuse of the engineering title and unlicensed practice of software engineering. The specific legal and regulatory environment in a jurisdiction will determine how and where this guidance should apply.

This paper also provides examples of software engineering in four practice areas, discussed in linked appendices to this document, as set out below:

- **APPENDIX A** – Construction: Structural and foundational analysis and design
- **APPENDIX B** – Manufacturing: Managing workplace risks through process safety
- **APPENDIX D** – Health care: Diagnostic imaging and medical image sharing
- **APPENDIX E** – Transportation: Public transit management systems

These examples are not intended to encompass the entire scope and depth of software engineering, but rather provide an interpretation of how software engineering applies in sample areas of practice.

## 3. USE OF THE TITLE "ENGINEER"

Irrespective of practices in other countries, representing oneself as an engineer in Canada is a protected act in Canada and requires licensure under the provincial and territorial engineering Acts.[1][2] Unlicensed individuals cannot use the title software engineer in their job titles, resumes, reports, letterhead, written and electronic correspondence, websites, social media, or anywhere else that may come to the attention of the public. Employers and individuals are not permitted to refer to themselves as a "Software Engineer" (or variations that the substitute the word "software" with a related discipline, for example: "Firmware Engineer," "Data Engineer," "Network Engineer", "Process Control Engineer", "DevOps engineer", and so forth) unless the individual is an engineer in the jurisdiction in question.

[1] An exception to this fact is the title "engineer" as it pertains to train operators.

[2] On July 19, 2022, all provincial and territorial engineering regulators in Canada, along with the CEO of Engineers Canada, signed a letter noting that "[u]se of 'software engineer', 'computer engineer' and related titles that prefix 'engineer' with IT-related disciplines and practices, is prohibited in all provinces and territories in Canada, unless the individual is licensed as an engineer by the applicable Provincial or Territorial engineering regulator." The full text of the letter is presented in Appendix F.

**4. PRACTICE OF SOFTWARE ENGINEERING**

According to the national definition in the Public guideline on the practice of engineering in Canada, engineers "plan, design, compose, evaluate, advise, report, direct or supervise, or manage any of the foregoing, work that requires the application of engineering principles and that concerns the safeguarding of life, health, property, economic interests, the public welfare or the environment." *Software* engineers are engineers whose area of practice is software or software-intensive systems, and so they would typically be involved in the specification, design, implementation, analysis, and validation of software. This includes the creation of software-based tools used to do other kinds engineering work. Software engineers' role in protecting the public is especially important in the case of software that is critical to the protection of life, health, or the environment, or in cases where software is substantially integrated into engineering works. A more detailed description of software engineers' work follows.

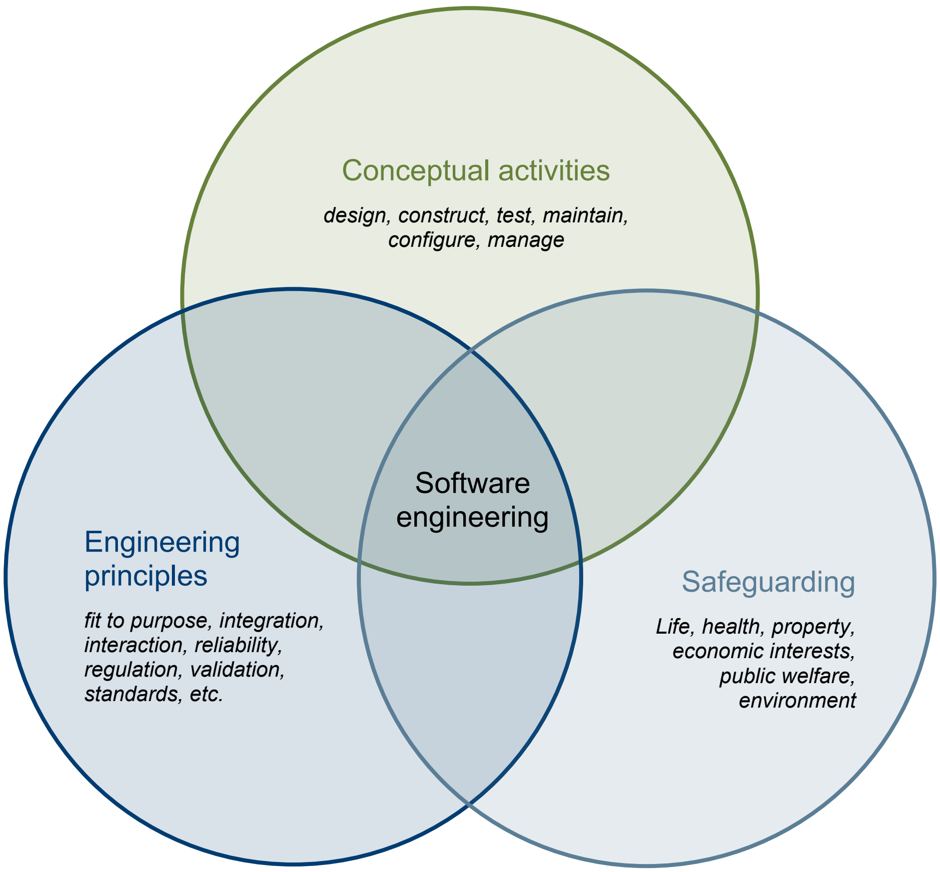


*Figure 1. Essential components of software engineering.*

**4.1 Software engineers use particular conceptual activities or combinations of them to do their engineering work**

Software engineers design, construct, test, maintain, configure, and manage software, and require understanding of the work's entire life cycle. In most jurisdictions, non-engineers, such as software developers, can perform these conceptual activities, providing that the work does not require the application of engineering principles, nor require the safeguarding of life, health, property, economic interests, the public welfare, or the environment. When the work requires engineering principles and safeguarding activities, non-engineers may only perform these conceptual activities as part of a team where an engineer whose area of practice is within the scope of the work takes responsibility for the overall system. In Quebec, software engineering can be carried out by non-engineers, providing they do not perform any activity reserved to engineers or carry themselves as engineers.[3]

**4.2 Software engineers apply engineering principles**

Throughout the lifecycle of software, software engineers:

- Consider constraints, design to best fit the purpose or service intended and address interdisciplinary impacts of the software on public welfare;
- Analyze technical and technology risks to the public and offer solutions to mitigate the risks;
- Apply engineering knowledge to design, implement, integrate, validate, deploy, and maintain software; and
- Understand solution techniques and independently verify the results of the software.

Determining if software engineering principles must be applied involves an assessment of the development process, the desired functions of the product, and the nature of the entities and environment with which it interacts.

The presence of some or all of the following indicate that engineering principles have been applied:

- Essential functions of the software cannot be fulsomely understood without thorough analysis of the behaviours of the software while interacting with:
    - Other software items;
    - Users;
    - Physical or environmental elements;
    - Elements already existing within the system (e.g. through regression testing)
- Essential functions of the software require interdisciplinary knowledge for core functions, to integrate with other software, or to integrate or interact with other systems or components
- The software requires integration with, and controls, regulates, or facilitates operation of, engineered work;
- The software requires understanding of the complexity around stringent requirements such as reliability, security, user interface, maintainability, testability, portability, interoperability;
- Availability, reliability, and/or maintainability are essential aspects of the software’s fitness for its intended application;

---

[3] Any member of L’Ordre des ingénieurs du Québec who practices software engineering is nevertheless under the jurisdiction of the regulator.

- Applicable regulatory or customer requirements mandate validation and verification of work at critical stages;
- Applicable regulatory or customer requirements dictate a higher standard of care and expectations of a structured and rigorous development process, often including adherence to published development process standards;
- Assessment of current and anticipated future software-related risks to the public or stakeholders could be substantially mitigated through the use of software engineering principles.

**4.3 Safeguarding life, health, property, economic interests, the public welfare, or the environment**

Engineers do not just provide technical solutions; they have an individual, professional responsibility for the impact of their designs, for the integration with other physical and virtual work with other disciplines, and for the completeness of the product. They are obligated to make ethical choices for the benefit of the public interest, even when in conflict with their individual interest.

If the engineering work requires the safeguarding of life, health, property, economic interests, the public welfare, or the environment, then a software engineer must take responsibility for the work.[4] Software that forms a significant or controlling part of essential public infrastructure, such as power distribution systems or essential public communications networks, or that controls systems that could put the public at risk, such as self-driving cars, clearly fits this definition, and so requires, in most jurisdictions, that a software engineer take responsibility for the system. In circumstances where the system is subject to demand-side legislation and requires safeguarding the public, for example in a power transmission system, only an engineer competent in software engineering can accept responsibility for the system's software components.

Indicators that software work involves the safeguarding of life, health, property, economic interests, the public welfare, or the environment include:

- **Deficient work could cause harm**: Deficient or missing function(s) of the software is likely to lead to property damage, harm to persons, or harm to the environment.
- **Deficient work could cause catastrophic damage**: Deficient or missing function(s) of the software, however unlikely, may lead to severe harm or death of persons, widespread economic consequences, or catastrophic damage to the environment;

[4] In Quebec, this only applies when the work has been designed by one holding the engineering title (i.e., ing.).

The risks to safety and public welfare are determined by the reasonably anticipated uses of the system in which the software is used. The requirement for sufficient safeguarding is implied by the nature, extent, and exposure of those risks with respect to stakeholders, including the general public, users of the system, and clients. In cases where safeguarding issues are potentially present, numerous factors must be taken into consideration by the software engineer:

- the current intended purpose of the work as well as reasonably foreseeable future applications over the life cycle of the software product
- all aspects of the development, deployment, and configuration processes for software associated with safety critical systems, the failure of which may cause or fail to prevent harm
- data protection, security, and infrastructure network operation as required to ensure the proper operation of the software
- the protection of public economic interests, beyond simply the interests of individual corporations or clients
- acknowledgement that the applications of computerized systems are constantly evolving, both in nature and degree, which leads to evolution of the public safety implications of such systems
- understanding the provenance of, assessing the appropriateness of, and defining a proper integration approach for third-party software libraries or components not designed or implemented by the software engineer.

Impact on public welfare goes beyond clients; it includes individuals and groups that might not have paid for services or products but would still suffer consequences should the software fail or is significantly deficient.

## 5. EMPLOYER OBLIGATIONS

Every firm engaged in the practice of engineering in Canada must register with the engineering regulator in the provincial and territorial jurisdiction(s) it practices (exceptions to this requirement exist in Québec). As discussed above in Section 3, representing oneself as an engineer in Canada (including in a job title) requires licensure with the provincial or territorial engineering legislation. Employers who have employees practising reserved software engineering activities must ensure that the employees are licensed with the applicable provincial or territorial engineering regulator, or that, alternately, an engineer directly supervises and takes responsibility for the unlicensed employee's engineering work.

## 6. BENEFITS OF HIRING AN ENGINEER

As is evident from Section 4, not all software practice is Software Engineering and so not everybody who develops software needs to be an engineer. Hiring an engineer gives several benefits for employers and the public, however. The engineering licensing system ensures that those who are licensed have been appropriately trained to undertake the work and have gained sufficient experience with appropriate mentorship to be qualified to take on the work independently. Engineers are also legally bound to adhere to their code of ethics and the relevant acts and bylaws of the jurisdiction in which they practice, which means, for example, that they will avoid any conflict of interest that may arise and will not undertake work for which they are not qualified. Above all, the obligation to hold the public interest paramount is the most important benefit to hiring an engineer.

**7. CONCLUSION**

The software field is broad and evolving, so it would be impossible to list all types of work that constitutes software engineering. The appendices following the main body of this paper provide samples of how software engineering, as defined by this paper, applies in different specialized areas of software practice. This paper's definition is intended to benefit regulators, who can use this information to inform their enforcement practices; individuals and employers, who may draw on it to inform their practises and identify software work that requires an engineer; and the general public, to properly seek out licensed engineers in instances where it is required by law.

## Appendices A-E – Software Engineering Practice Areas

### A note on the appendices

From safety-critical functions in industrial manufacturing or construction to integrated diagnostic tools in health care or for the optimization in transportation, software engineering plays a variety of key roles involving analysis, specification, design, development, certification, maintenance and testing of software systems. It may occur in both established engineering disciplines and in circumstances that are not traditionally engineering (e.g. finance, communications, information management). Regardless of the context in which software engineering occurs, software engineers are bound by their jurisdictions' codes of ethics.

The sample areas of practice in the following appendices are not intended to describe the entire scope and depth of software engineering within industries, but rather serve as illustrations of ways that software engineering, as defined by this paper, applies in a variety of contexts.

Please consult individual provincial and territorial engineering regulators' acts to learn about the legal requirements applicable in each jurisdiction.

- **APPENDIX A** – Construction: Structural and foundational analysis and design
- **APPENDIX B** – Manufacturing: Managing workplace risks through process safety
- **APPENDIX C** – Health care: Diagnostic imaging and medical image sharing
- **APPENDIX D** – Transportation: Public transit management systems
- **APPENDIX E** – Letter from Canadian engineering regulators on "Use of 'Software Engineer' and Related Titles in Canada"

The design, maintenance, and operations of systems that do not require the three elements of software engineering, as outlined in this paper, fall outside of the scope of regulation. The bulk of an engineer's software activities may consist of non-engineering work. Non-engineering software work may include, but is not limited to, development of non-safety-critical software functions (e.g. saving and printing functions in a software suite) or front-end application design (e.g. UX design, visual elements).

## PRACTICE AREA A: CONSTRUCTION

**Focus: Structural and foundational analysis and design**

**Background**

As evidenced by a number of structural failures recorded in the news and historical records, construction projects can be risky and complicated ventures. Today, projects often involve curved structural elements, a wide array of loads and deflections to consider, as well as the seamless integration of existing infrastructure with new structures. Among other factors, this complex work requires striking a balance among efficient design processes, structural performance, and the initial architectural vision. In this environment, structural and geotechnical engineers are constantly being pushed to become more efficient in the way they work, particularly by increasing the speed, quality, and accuracy of their designs. One method for achieving this is to develop software to perform the more repetitive, labour-intensive, and error-prone tasks that were traditionally completed manually.

***Safeguarding***

Construction engineering software allows the creation of models and simulations that factor in significantly more information than manual designs, enabling increased complexity and reliability in the engineering team's designs. To effectively develop such software, whether for a custom application or off-the-shelf product, a designer must be able to solicit a robust understanding of client needs, and must have sufficient knowledge to be able to test and validate the software's results. Additionally, the software must be developed in a way that anticipates reasonably foreseeable risks associated with its future uses. While ultimately the engineering firm assumes responsibility for design and construction of a project itself, the risks associated with the ineffective design of construction engineering software still involve the ***safeguarding of life, health, and property.*** As a result, the design and analysis software must be designed by a software engineer.

***Conceptual Activities and Engineering Principles***

Applications used to support construction engineering projects must be designed in such a way that they safely and reliably meet client requirements. To do this effectively requires knowledge of **engineering principles**, including the ability to consider constraints and design in a way that is fit to purpose. Notably, an application itself may only minimally require technical engineering expertise (e.g. mathematical calculation), but only someone with knowledge of engineering principles will be able to safely gather requirements and validate whether they have been properly satisfied.

Within this context, safe and reliable software design requires **conceptual activities**, which entail systematic design, construction, testing, maintenance, configuration, and management of the software. Moreover, effective development requires understanding of the work's entire life cycle. When there is a risk that a lack of proper conceptualization may cause public harm, the work must be undertaken by a software engineer.

**Example: Computational mechanics**

The use of computational mechanics in project design—the intersection of mechanics, applied mathematics, and computer science—has generated considerable efficiencies for the construction industry. To safeguard the public, systems employing computational mechanics must be designed or overseen by software engineers, because the proper development of such systems requires conceptual activities (e.g. systematic development and validation) and the application of engineering principles (e.g. thorough analysis of the required software capabilities and functions).

For example, computed parametric modelling can be used to determine appropriate structural configurations and geotechnical properties to enhance the seismic performance of a bridge. For this purpose, software models are created, for example, to analyze a set of ground motions with various intensities, including dynamic soil-bridge interaction effects. In the analyses, the effect of various structural and geotechnical properties (e.g., foundation soil stiffness, backfill compaction level, pile size and orientation, abutment height, and thickness) are considered. Such modelling can be both more computationally efficient and, often, more reliable than manual modelling methods. However, its safe development requires an engineer's perspective, because an engineer will have the broader knowledge to validate its outputs.

**Example: Building information modelling**

Building information modelling (BIM) has become a valuable tool for managing data and generating designs during the entire lifecycle of a project. BIM applications allow for the integration of multi-disciplinary data, which can be used to model simulations that provide engineering teams with critical information to assist in their decision making and planning (e.g. pre-fabrication potential, waste management and mitigation, required materials, design conflicts, etc.). Such software has the potential to help teams identify and mitigate design and/or safety issues prior to actual construction. While a software engineer is not required to develop all aspects of BIM applications (e.g. the application's interface), and while accountability for construction decisions ultimately resides with the construction firm, engineering expertise is required to oversee and validate any functionality that has the potential to impact safeguarding of the public and its welfare. This is particularly the case where the construction firm may not have knowledge of the application's back-end functionality.

**Conclusion**

Much of construction software design and development (e.g. the software's interface, data entry support, saving and printing functionality, etc.) does not require the expertise of an engineer. It is only in cases where an element of the application has the potential to impact safeguarding of public welfare, and where such risks can be mitigated with an expert knowledge of engineering principles and conceptual activities, that a software engineer must undertake or validate/oversee the development of software.

**PRACTICE AREA B: MANUFACTURING**

**Focus: Managing workplace risks through safe software engineering**

**Background**

Today's manufacturers face operational challenges surrounding the safety of personnel, equipment, and the environment. They must meet standards set out by industry bodies, regulation, and legislation. Process safety allows for the management of these factors through the implementation of controls to reduce risks such as high-impact fires, explosions, accidental chemical releases, structural collapses, equipment malfunctions, corrosion, component failures, and disruptions. When these risks are mismanaged, the consequences can be catastrophic.

To address risks and improve performance in manufacturing systems process control systems and safety systems (both outlined below) are often merged into a one platform, called integrated control and safety systems (ICSS). These systems, which use both hardware and software to increase automation of industrial functions, provide integrated tools that can help reduce the need for labour, increase reliability, and decrease the possibility of systematic errors. The nature of these systems means that the software used to manage and automate these systems must be designed with an understanding of how different physical and computing components will interact, and must be designed with extensive, effective cybersecurity measures.

***Conceptual Activities and Engineering Principles***

When designing and implementing software systems, software engineers working in manufacturing ***apply engineering principles*** by drawing on their knowledge of and integrating existing, new, and emerging technologies. The software systems they design must reliably process real-time information and must automatically manage control systems relating to manufacturing processes. This requires the ability to test and analyze the way integrated systems will interact, and a knowledge of the larger systems' entire lifecycle, including the impacts of its component technologies' obsolescence and/or the potential impacts of incorporating new components with the system in the future.

In terms of ***conceptual activities***, software engineers working in manufacturing use their understanding and operation of the development stages—analysis, design, development, testing, and maintenance. The software engineer engages in these conceptual activities to ensure, for example:

- Responsiveness of software systems to the facility's process automation systems
- Effective quality management and quality assurance
- Reduced energy costs and overall enhanced performance of systems within safe parameters
- Accurate real-time information transfer and satisfaction of data visibility requirements to ensure safe monitoring
- Peak asset performance by creating software systems that can monitor connected smart instrumentation and field devices, and manage calibration schedules and equipment maintenance
- Software systems that allow for successful migration to new human-machine interfaces
- The capability of a software system to be reconfigured or adjusted for future changes to safety guidelines or other required parameter changes

***Safeguarding***

Numerous dimensions of software work in manufacturing involve ***safeguarding life, health, property, economic interests, the public welfare, and the environment***. Software engineers have the knowledge needed to design software that can integrate and manage disparate manufacturing systems, based on specific criteria, for safe and effective operation. This safeguarding includes cybersecurity protection to ensure, for instance, that unauthorized configuration changes to a safety system cannot be made, and to prevent interference with the system's ability to accurately represent the status of the system's instrumentation (e.g., the loss of alarms, total loss of visibility, spoofing the operator). Software engineers also ensure that software systems safely separate control and safety components, so that an intrusion into the basic control system cannot circumvent safety instrumentation systems. In a properly engineered software control system, in other words, the failure of a non-safety-related function should not be able to cause a failure of a safety-related function.

Because manufacturing processes are increasingly managed through software systems, and because these systems require expert knowledge of the way that systems must be designed and integrated with others for their safe functioning, a software engineer is required for their design.

**Example: Industrial control systems**

Expertise in software engineering is essential in the development and implementation of software used to manage and automate the *industrial control systems* used extensively in industries such as chemical processing, pulp and paper manufacturing, power generation, oil and gas processing, and telecommunications. Software used to manage industrial control systems must automatically receive and integrate data from remote sensors measuring process variables (e.g., pressure, temperature, level, flow), compare the collected data with desired setpoints, and derive command functions that are used to control a process through the final control elements, such as control valves. This software must also be able to safely and reliably communicate with the components it monitors and manages, ranging from a few modular panel-mounted controllers to large interconnected and interactive distributed control systems (DCSs) with many thousands of field connections. To effectively manage, process, and adjust controls based on the data coming from such varied inputs, the software's design requires understanding of the interaction of the software with physical systems, the interaction of the software with hardware and other embedded systems, and the lifecycle, limitations, and risks of such systems.

Due to the numerous, complex elements that must be seamlessly integrated into these software systems, and the safeguarding impacts that their failure can entail, they should be designed by a software engineer.

**Example: Industrial safety systems**

*Industrial safety systems* are designed to protect people, facilities, and the environment by putting in place measures that can automatically respond when processes go beyond allowed control margins. They are crucial in all potentially hazardous plants, such as oil and gas and nuclear plants. When such a system relies on a software component, such as when the processing of data points from within a gas plant is automated to trigger a safety measure when the data exceeds certain temperature parameters, the software must be designed by a software engineer. This is because a software engineer will have knowledge of the conceptual activities required for safe design of such systems, including knowledge of how to determine methods for testing and maintaining the systems so as to ensure their safe operation. A software engineer also will understand the interaction of the software system with other physical and hardware systems, and will be able to take measures to secure the system against cyberattacks (in this case, it may just be a matter of identifying the risk and ensuring a cybersecurity expert is consulted in the plant's network design).

**Conclusion**

Like any engineer, a software engineer is duty-bound to develop products and systems that "Hold paramount the safety, health and welfare of the public and the protection of the environment and promote health and safety within the workplace."[5] In the case of manufacturing, software engineers must define appropriate requirements, define operating use cases, and develop reliable software systems, drawing on their broader knowledge of legislation, physical systems, engineered hardware, and software security. Due to the centrality of these systems in manufacturing safety, and the catastrophic consequences of error, only an engineer competent in software engineering should be tasked with their design and implementation.

[5] https://engineerscanada.ca/publications/public-guideline-on-the-code-of-ethics, accessed January 24, 2021.

**PRACTICE AREA C: HEALTH CARE**

**Focus: Diagnostic imaging and medical image sharing**

In Canada's complex and fragmented health care system, there has been widespread recognition of the role that IT and software-based solutions can play in delivering high-quality patient-centred care. Health care professionals are continuously seeking out ways to achieve better outcomes with minimal resources, needing, for example, to share data instantaneously across multi-disciplinary teams in various locations. Within this ecosystem, medical technologies and systems often require complex software integrations that both comply with legal and regulatory frameworks and that are designed with safeguarding of the public at the forefront.

***Conceptual Activities and Engineering Principles***

Software engineers develop and oversee the development of software systems that assist in medical image processing, secure medical recordkeeping and sharing, patient diagnosis, patient monitoring, and clinical decision-making. A software engineer considers the entire lifecycle of the component, independently verifying its results and analyzing risks to its interaction with other systems, including physical systems, medical hardware, and other software.

In the field of health care, a software engineer may undertake a variety of engineering activities relating to ***conceptual activities*** and the ***application of engineering principles***, including:

- development of a software component that processes imaging information in a manner that accounts for its entire lifecycle (i.e., through design, construction, testing, maintenance, configuration, scalability, availability, and management)
- designing software components of monitoring equipment in such a way that they operate within constraints pertaining to accuracy, latency, and execution time of algorithms
- ensuring compliance of software for medical information management in such a way that it meets requirements established within federal and/or provincial and territorial information protection and privacy laws
- design and development of imaging and information sharing software in such a way that it minimizes risks relating to data breaches and cyberattacks
- creating software interfaces for medical equipment in such a way that they accurately and reliably represent medical data, developing safeguards to ensure the integrity of information in various use cases (e.g., power shortages, network failures, data corruption, cyberattacks)
- ensuring secure transmission of data in automated of medical record keeping systems
- providing quality assurance pertaining to software, user interfaces, systems interoperability, process automation, etc.
- designing the software so it may eventually be extended and further refined, in accordance with sustainable development. This also means carrying out a technological watch so the software remains secured throughout it's lifecycle (e.g., new cyberattacks, dependant software vulnerabilities)
- recommend deployment strategies that take security, scalability, availability and resources consumption into account.
- recommend technologies, deployment strategies, updates, fixes

In medical fields, software engineers are often required for work in areas where broad interdisciplinary engineering expertise is required (e.g., robotics, mechatronics, virtual and augmented reality, mobile, wearable and implantable devices, health informatics). Their interdisciplinary knowledge and expertise on the interaction of systems is required to ensure safe and effective integration of software components with other systems.

***Safeguarding***

Doctors hold the ultimate responsibility for medical decisions, but accurate and reliable recording, representation, storage, and transmission of patient information is critical to medical professionals' ability to properly conduct their work. Improperly designed and implemented software in medical care environments can have serious repercussions, including incomplete or biased information, increased risk of misdiagnoses, and even loss of patient life[6]. Even in cases of medical data breaches, which pose minimal immediate risk to patient's life, medical information is considered highly personal and receives extensive legal protections.

Because of these risks, and because of the extensive knowledge required to properly mitigate them, software engineers must develop or oversee the development of software components involved with medical equipment and medical information management systems. Engineers are bound by their code of ethics to ***safeguard life, health, and public welfare***, and so working with engineers over the lifecycle of software components for these systems helps medical professionals ensure that they are meeting their own professional responsibilities and requirements.

**Example: Rapid diagnosis for a stroke patient**

For a patient who has experienced a cerebrovascular accident (CVA), or stroke, fast, accurate treatment is critical. In the diagnostic phase, medical staff will order imaging tests to rule out other conditions and ultimately determine which type of stroke a patient has had. Immediate treatment may minimize the long-term effects of a stroke and even prevent death. Certain types of strokes, for example, can be treated with a dissolving agent if it is administered within approximately three hours of the event.

Using a conventional process, images of the patient would be manually transferred to the imaging system, where they could be searched, selected, and uploaded for analysis. The clinician would analyze and save the images, prepare the results, and subsequently transfer them to the physician for interpretation. In a system where minutes can mean the difference between life and death, automated solutions for the transmission of information offer a valuable prospect for improving patient outcomes.

A software engineer is trained to implement solutions that consider risks such as improper data management and transmission. If, for example, data status is not properly monitored and communicated to the software user, a doctor may be making decisions based on inaccurate information. Likewise, if patient information is stored or transmitted using improper techniques, it could be susceptible to data corruption and/or malicious actions. As engineers are duty bound to have understanding of the software's function within its deployed context, and an understanding of its vulnerabilities in relation to other systems, including physical systems, they can mitigate against a wide range of risks that a specialist such as a cybersecurity expert would not necessarily perceive. Given these factors and numerous others relating to safeguarding the public, software engineers are required for the design of safe, properly implemented medical information software systems.

---

[6] A well known example of the risks of improper software engineering in medical technology is Therac-25, a computer controlled radiation therapy machine whose flawed software system design periodically led to patients receiving dangerously high, and in some cases lethal, doses of radiation.

https://escholarship.org/uc/item/5dr206s3, accessed June 17, 2022.

**Example: Automated workflows**

The previous example illustrated the life-saving benefits of software-based tools to improve the speed with which information can be shared. Another key area where software has been beneficial to improving efficiency in the medical profession is in the automated analysis and rendering of data to augment decision making.

With the use of artificial intelligence-assisted automated workflows, information is integrated and prioritized automatically. In the example of patient imaging, when the patient undergoes the imaging procedure, a “zero-click” automated platform could be used to perform the following functions:

- Process and categorize images through innovative imaging modalities;
- Analyze and detect abnormality through the integration of advanced visualization algorithms, helping detect, classify, and characterize conditions at the point of image acquisition;
- Prioritize and issue alerts on results, which are based on integrated data that can be shared across all networks (regardless of imaging vendor or system).

Such an automated workflow enables efficient sharing of information allowing for the faster transmission of results for interpretation by the physician. When properly designed and implemented, such a system has the potential to increase overall productivity and responsiveness of medical facilities, reduce manual burden and human error, and ultimately improve the patient’s prospects of a favourable outcome. Collaboration between software engineers and healthcare professionals is of the utmost importance at that stage, so that system generated results may be compared to real-life results. Biases may be discovered and corrected using that approach.

In the design of such a system, the application of engineering principles are required in several ways. The designer must consider hardware and physical constraints and design the automation software in such a manner that it properly interacts with all integrated systems, including, for example, medical information sharing systems. This requires not only expertise in design, but also expertise in testing and quality assurance, regulatory compliance, and maintenance and management of the system over its lifecycle. The system must be designed in such a way that it considers the future integration of other systems or the obsolescence of other components with which it is integrated. Finally, and most critically, it must employ safeguards and measures that support the many cases in which a medical professional is required to further assess diagnostic information. While automated systems can be enormously beneficial to the fast and accurate diagnoses of patients, the numerous risks involved with these systems require a software engineer to ensure they are being designed in a safe manner.

**Conclusion**

Medical applications of software engineering are numerous, and center around the secure and accurate measurement, recording, processing, and sharing of patient data. Engineered software in this area has the potential to reduce administrative burdens and increase the speed of diagnoses, but it must be developed according to engineering principles to ensure medical professionals can accurately and reliably conduct their work. As these applications have involve ***safeguarding of life, health, and public welfare*** they must be developed and implemented by software engineers.

## PRACTICE AREA D: TRANSPORTATION

**Focus: Public transit management system**

With public transit being transformed from a system of static, scheduled, fixed routes to dynamic on-demand networks, the demand for real-time measurement and analysis of transit systems has become a necessity. Transit authorities receive real-time information which allows for effective management of fleet resources. These systems can track transit routes and ridership, as well as providing live reporting of maintenance issues, breakdowns, delays, and other incidents. The systems also incorporate safety features such as event-triggered video/audio feeds and push-button calls to emergency services.[7]

Given that disruption of transit services could have widespread impacts of the functioning of a city and its infrastructure, including significant economic impacts and a rise in traffic incidents, systems must be designed in a way that ensures secure, efficient interactions among a wide variety of components (e.g., GPS systems, dispatching services, real-time ridership data, traffic information services, route disruption data sources, safety-event triggers). Software plays a critical role in the analysis and representation of these numerous components, and because of the system's overall complexity, the software's design requires the application of conceptual activities and engineering principles. Software engineers' understanding of all aspects of the software development life cycle, including coding standards, code reviews, source control management, build-processes, testing, and operations are required for the safe design of public transit management systems.

### *Conceptual Activities and Engineering Principles*

Whether its in developing algorithms to process real-time information from numerous transit data sources, or developing automated dispatching systems to manage emergencies, software engineers involved with the design and development of PTMSs undertake a wide variety of ***conceptual activities*** and the ***application of engineering principles***, including:

- Designing, prototyping, testing, and implementing complex, highly scalable, reliable systems and web-based applications to ensure that they are fault-tolerant and reliable under a wide variety of use cases
- Processing, integrating, and representing data from a variety of sources and platforms, including physical systems (e.g. buses, roads), in such a way that takes into consideration and mitigates risks such as faulty instrumentation, tampering, or corrupt data.
- Delivery of accurate transit data such as schedules and delays to transit customers, integrating disparate data sources for a variety of use cases while anticipating software-related risks such as DDoS attacks or hacking
- Acquisition, storage, and management of ridership data (e.g., transit passes, customer accounts) in such a way that it complies with privacy regulation and other laws relating to cybersecurity and data management
- Modelling complex interactions in a real-time system using efficient data structures that support millions of transactions per minute by users and operators;

[7] For more information on the topic of artificial intelligence engineering technology in autonomous and connected vehicles, please consult Engineers Canada's national position statement on this topic. A recent example illustrating risks in the area of vehicle automation can be found in case studies of the Boeing 737 Max's Maneuvering Characteristics Augmentation System (MCAS), whose faulty implementation led to two crashes between 2018 and 2019 and the subsequent grounding of the 737 Max fleet for 20 months.

- Integrating operating systems, memory management, performance/resource optimizations, database interactions, network programming, concurrency, multithreading, fault tolerance, and the monitoring, security, and operability of the system;
- Putting in place extensive testing, verification, monitoring, and validation protocols, which requires an advanced understanding of quality assurance principles.

***Safeguarding***

When designing PTMSs and other transit-related systems, software engineers must often integrate multiple hardware and software systems to ensure their safe and secure interoperation. If these systems are not designed in such a way that accounts for, among other factors, their interaction with other systems, their full life cycle from development to obsolescence, cybersecurity risks, and any potential risks to the public, they can pose substantial danger to the safety of users (including operators) and the functioning of city infrastructure. To ensure their safety, these software systems must be designed in such a way that they utilize the expertise of interdisciplinary teams comprised of planners, transit engineers, operators, and other relevant professionals; a software engineer's knowledge of others' expertise is critical to its safe and proper integration within software systems. For the design of any software components of PTMSs that could involve risks involving ***safeguarding life, health, and public welfare***, such as those outlined here, a software engineer is required.

**Example: Public transit management systems**

Public transit management systems (PTMSs) are advanced software applications or suites that utilize and integrate multiple applications and data sources to assist in the management of public transit systems. To support the immense task of providing transit information and tracking routes—and in some cases, acting as a transactional platform—PTMSs often integrate multiple management systems such as accounting software, payment processing software, and route scheduling tools. To accomplish these tasks, a PTMS must be designed in such a way that takes into consideration the lifecycle of system components and vulnerabilities to cybersecurity attacks, while remaining reliable and responsive with regard to its intended application. A software engineer provides a higher standard of care and expectations of a structured and rigorous development process, often including adherence to published development process standards. In short, because of the complexity of a PTMS and the multiple factors involved in its development and management, to be designed safely, it requires ***conceptual activities*** and the ***application of engineering principles.***

In addition to assisting with the management of transit system resources, a PTMS may also integrate features to support incident investigations, such as driver assaults, traffic accidents, or fraudulent activity. These features must be designed in such a way that ensures accurate, reliable, and regulatory-compliant recording and monitoring of user data (e.g. operator behaviour, vehicular performance, registered rider sign-ins). For example, when questions arise regarding a preventable traffic accident, a PTMS will have collected data, such as information regarding the vehicle's geographical location, speed, and driving direction, at regular intervals of time. Because of the risks involved with faulty or improperly managed data, when a PTMS software component incorporates functions involving ***safeguarding of life, health, property, economic interest, public welfare, and the environment***, it must be designed by a software engineer.

**Conclusion**

Transit systems and city infrastructure are complex entities with an immense number of continuously changing data points. When properly designed, the integrated systems comprising a PTMS can help manage complexity, enhance transportation and safety, and safeguard against the significant problems that can occur in relation to service disruptions. Due to their complexity and due to the risks involved with improperly conceptualized or integrated PTMSs, a software engineer should undertake or oversee their development, and must be involved with any components relating to safeguarding the public.

**BIBLIOGRAPHY**


1. Canadian Engineering Qualifications Board, *Software Engineering Syllabus*, 2004. http://www.engineerscanada.ca/e/pu_syllabus_1.cfm
2. Institute of Electrical and Electronics Engineers, *Guide to the Software Engineering Body of Knowledge* (SWEBOK), 2004. http://www.computer.org/portal/web/swebok/2004guide
3. Institute of Electrical and Electronic Engineers, *IEEE Standard Glossary of Software Engineering Terminology,* IEEE Std 610.12-1990, New York, NY. http://standards.ieee.org/findstds/standard/610.12-1990.html
4. Institute of Electrical and Electronic Engineers, *IEEE Standard for Software Safety Plans,* IEEE Standard 1228-1994, New York, NY. http://standards.ieee.org/findstds/standard/1228-1994.html
5. International Atomic Energy Agency, *Verification and validation of software related to nuclear power plant instrumentation and control,* Technical reports series no. 384, Vienna, 1999. http://www.iaea.org/NuclearPower/IandC/index.html
6. International Atomic Energy Agency, *Software for computer based systems important to safety in nuclear power plants : safety guide,* Safety Standards Series No. NS-G-1.1, Vienna, 2000. http://www.iaea.org/NuclearPower/IandC/index.html
7. Biomedical Engineering Desk Reference, Buddy D. Ratner et al, Academic Press, 2009, ISBN: 9780123746467
8. Heath Canada, Notice - Software Regulated as a Class I or Class II Medical Device, December 3, 2010. http://www.hc-sc.gc.ca/dhp-mps/md-im/activit/announce-annonce/md_notice_software_im_avis_logicels-eng.php
9. Food and Drug Administration, Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices, May 11, 2005. http://www.fda.gov/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/ucm089543.htm
10. For an illustration of the wide range of ways computer systems can be detrimental to public welfare see the Association of Computing Machinery's Forum on Risks to the Public in Computers and Related Systems (http://catless.ncl.ac.uk/Risks/).

## Appendix E: Letter from Canadian engineering regulators on "Use of 'Software Engineer' and Related Titles in Canada"

In July 2022, Engineers Canada and the 12 engineering regulators across Canada co-signed a statement reiterating that the use of titles such as "software engineer", "computer engineer", and similar titles that prefix "engineer" within IT-related disciplines and practices are restricted to those who are licensed as an engineer.

Read the statement here.