Natural Sciences and Engineering Research Council of Canada | Conseil de recherches en sciences naturelles et en génie du Canada

Institutional Identifier	FORM 101 Application for a Grant PART I	Date
System-ID (for NSERC use only) 341042327		2015/06/03

Family name of applicant	Given name	Initial(s) of all given names	Personal identification no. (PIN)
Rigby	Peter	PC	**Valid** 309207

Department	Institution that will administer the grant
Computer Science and Software Engineering	Concordia

Language of application	Time (in hours per month) to be devoted to the proposed research / activity
[X] English [ ] French	25

Type of grant applied for	For Strategic Projects, indicate the Target Area and the Research Topic; for Strategic Networks indicate the Target Area.
Engage Grants for universities	

Title of proposal
Test Prioritization and Localization at Ericsson

Provide a maximum of 10 key words that describe this proposal. Use commas to separate them.
Empirical Software Engineering, Software Analytics, Quality Assurance Practices, Test Prioritization, Fault Localization, Statistical Models, Software Development Work Practices

Research subject code(s)		Area of application code(s)	
Primary	Secondary	Primary	Secondary
2706	2705	801	802

CERTIFICATION/REQUIREMENTS

If this proposal involves any of the following, check the box(es) and submit the protocol to the university or college's certification committee.

Research involving : Humans [ ] Human pluripotent stem cells [ ] Animals [ ] Biohazards [ ]

Indicate if the proposed research takes place outdoors and if you answered YES to a), b) or c) – Appendix A (Form 101) must be completed

[X] NO [ ] YES

TOTAL AMOUNT REQUESTED FROM NSERC

Year 1	Year 2	Year 3	Year 4	Year 5
25,000	0	0	0	0

I certify that this project will involve only industry partners with whom no prior research partnership has taken place: Yes

SIGNATURES (Refer to instructions "What do signatures mean?")

It is agreed that the general conditions governing grants as outlined in the NSERC *Program Guide for Professors* apply to any grant made pursuant to this application and are hereby accepted by the applicant and the applicant's employing institution.

Applicant

Applicant's department, institution, tel. and fax nos., and e-mail

Computer Science and Software Engineering
Concordia
Tel.: (514) 8482424 ext. 3167
peter.rigby@concordia.ca

Head of department

Dean of faculty

President of institution (or representative)

The information collected on this form and appendices will be stored in the Personal Information Bank for the appropriate program.

PROTECTED WHEN COMPLETED

Version française disponible

Canada

Personal identification no. (PIN)	Family name of applicant
Valid 309207	Rigby

SUMMARY OF PROPOSAL FOR PUBLIC RELEASE (Use plain language.)

This plain language summary will be available to the public if your proposal is funded. Although it is not mandatory, you may choose to include your business telephone number and/or your e-mail address to facilitate contact with the public and the media about your research.

Business telephone no. (optional):

E-mail address (optional): peter.rigby@concordia.ca

Ericsson develops and supplies equipment, software, and services for global mobile networks. Ericsson in Ottawa develops cellular base stations that are the interface between cell phones to the physical backbone networks. This high reliability environment requires testing on expensive hardware that can simulate cell phones interacting with base stations. Some tests are expensive running for days, while others take seconds. When a test fails, the software integration process stops and an engineer must intervene. Our goal is combine test prioritization with empirical software engineering to mine the historical artifacts including past code changes, bugs, peer reviews, and test runs to create statistical models to (1) prioritize tests and (2) help developers categorize and locate the cause of a failure.

Research into test prioritization work has been limited by information about past runs and links to other development artifacts. Ericsson collects massive histories of logs, test runs, and development artifacts, in part to provide for reproducibility of software and auditing. Much of the current analysis is done manually where developers must sift through this information to locate the root cause of a failure.

My research team has extensive experience in mining and linking software artifacts. Combined with Ericsson's rich data store, we plan to create statistical models and classifiers to efficiently order tests, and to relate other artifacts to provide additional information about the root cause of failures and the factors that contribute to failures. For example, we may find that a certain set of short running tests find over 70% of all failures and that a change to a module that controls signaling by an inexperienced developer will lead to an 10% increase in the likelihood of a high impact failure. These outcomes would increase the productivity of Ericsson Ottawa's 1000 employees and advance the state-of-the-art in understanding which historical measures help in test prioritization and localization.

Other Language Version of Summary (optional).

Personal identification no. (PIN)	Family name of applicant
Valid 309207	Rigby

See instructions for further details.

PROPOSED EXPENDITURES

	Cash	In-kind
1) Salaries and benefits		
a) Students	24,500	
b) Postdoctoral fellows	0	
c) Technical/professional assistants	0	15,000
d)	0	
2) Equipment or facility		
a) Purchase or rental	0	0
b) Operation and maintenance costs	0	
c) User fees	0	0
d)	0	
3) Materials and supplies		
a)	0	0
b)	0	
c)	0	0
4) Travel		
a) Conferences		
b) Field work	0	0
c) Project-related travel	500	
d)	0	
5) Dissemination		
a) Publication costs	0	
b)	0	
6) Technology transfer activities		
a) Field trials	0	
b) Prototypes	0	
c)	0	
TOTAL PROPOSED EXPENDITURES	25,000	
Total support from industry	0	
Total support from university		
Total support from other sources		
AMOUNT REQUESTED FROM NSERC	25,000	

PROTECTED WHEN COMPLETED

Personal identification no. (PIN)	Family name of applicant
Valid 309207	Rigby

Supporting organizations are not required to make cash or in-kind contributions for this grant. However, if there are any contributions, please report them in the following table, and describe any in-kind contributions provided in the budget justification.

Name of supporting organization

Ericsson Canada

CONTRIBUTIONS FROM SUPPORTING ORGANIZATIONS	
Cash contributions to direct costs of research (Transfer amounts to page three (3); except those for the Ship Time program.)	0
In-kind contributions to direct costs of research	
1) Salaries for scientific and technical staff	15,000
2) Donation of equipment, software	0
3) Donation of material	0
4) Field work logistics	0
5) Provision of services	0
6)	0
Total of in-kind contributions to direct costs of research	15,000
In-kind contributions to indirect costs of research (not leveraged)	
1) Use of organization's facilities	0
2) Salaries of managerial and administrative staff	0
3)	0
Total of all in-kind contributions	15,000
Contribution to postsecondary institution overhead	0

Contributions from Supporting Organization

Ericsson Ottawa's in-kind contribution will be substantial. The main point of contact will be Griffiths, who is the Section Manager for the Development Environment in Ottawa. Engineers on Griffiths's team will have to introduce the research team, especially the on-site masters student, to the test and development processes. They will also help the masters student to extract appropriate attributes and interpret them in context. Rigby and Rahman will visit Ericsson at least twice a month to meet with Griffiths's team and to help the masters student with data mining and modelling. Further, McKenna, the lead test Engineer, will help in evaluating the results and ensuring that they have an impact on Ericsson's continuous integration environment.

Over the six month period of the grant, we expect the teams to contribute an average of 12 hours per week. In total, the in-kind contribution will be $15,000.

Budget Justification

We plan to support one PhD student and one masters student with this grant. The students will be under the direct supervision of Dr. Rigby. Both student will be supported for 6 months by the proposed grant, with $11,500 and $13,000 going to the masters and PhD student, respectively. If the proposed grant is successful, we will discuss a MITACS grant with Ericsson to continue this work.

As discussed in the proposal, the recruited masters student will work directly onsite during this grant to mine, link, and model Ericsson data.

One of Rigby's PhD students, Rahman, will investigate test prioritization in the context of release engineering. As we discussed in the proposal, releases can be slowed down by repeated test runs and failures. Rahman's PhD topic is in release engineering, and a chapter on test prioritization would make an interesting novel addition in his thesis. It would also expose him to interesting new datasets and methodologies used by Ericsson. There is no expectation of intellectual property issues as the findings involve the Ericsson development environment instead of Ericsson products.

We will need a small budget to travel to Ericsson Ottawa for meetings ($500).

Relationship To Other Research Support

My research area is empirical software engineering and the goal of my ongoing research program is to understand how and help developers to produce successful software systems. I am also interested in statistical machine translations and summaries of code identifiers using freeform text, such as StackOverflow documentation.

The main goal of my NSERC Discovery grant, Contemporary Software Peer Review: Modern practices, fault prediction, and extraction of design decisions is to understand the software review practices in industrial firms as opposed to open source projects I studied in my thesis (from this grant, I have published a paper at the top tier conference Foundations of Software Engineering). In the proposed Engage grant, I will be able to use my expertise and scripts to extract Ericsson data. A possible synergy between the grants, would be to use peer review data in models for test prioritization.

My Department of National Defense (DND), NSERC, industry (KDM Analytics) grant, The Impact of Disruptive Events on Software Systems, involves studying disruptive events that lead to poor software outcomes. A disruptive event, such as developer turnover, will have risks and mitigating factors that we are measuring. While there is no direct overlap between the grants, it may be possible to observe how test failures disrupt the software development and release processes.

The goal of my FRQNT grant is to translate code elements, classes and methods, from English into French. This grant does not overlap with the Engage proposal. The identifier names (eg names of classes and methods) of most major software libraries are based on English terms that capture the purpose of identifiers (eg the Android 'AccountManager' class). There are thousands of identifiers on the Android project, putting non-English speakers at a disadvantage. Statistical translations (eg Google Translate) of technical documents using non-technical language models, results in incorrect translations of technical terms (eg the term 'Window' has a technical meaning quite different from its non-technical meaning). Unlike non-technical document translation, we have observed that library identifiers are not translated when used in multiple languages, 'AccountManager.addAccountExplicitly qu'est-ce que cette fonction fait?' I am in a unique position to create language models to translate software documents because, as a postdoctoral research, I developed a technique that extracts identifier names in freeform text and code fragments. We can take two comparable corpora, such as, the community forums on Android in French and English and statistically determine which words tend to co-occur with each identifier. These co-occurring words represent a language model that describes each identifier in its respective language, which when aligned, allows us to 1) describe the purpose and behaviour of an identifier 2) increase the quality of translations of entire documents that discuss identifiers.

Test Prioritization and Localization at Ericsson

Overview of the Company-Specific Problem

A cellular base station connects cell phones to a voice and data network via LTE (4G) or 3G standards. The software that runs on these base stations contains not only complex signalling logic with stringent real-time constraints, but also must be highly reliable, providing safety critical services, such as 911 calling. At Ericsson Ottawa, testing base station software is time consuming and involves expensive specialized hardware. For example, testers may need to simulate cellular devices, such as when a base station is overwhelmed by requests from cell users at a music concert. In order to maximize the return on investment of Ericsson's testing efforts, we plan to use historical artifacts including past code changes, bugs, peer reviews, and test runs to create statistical models to (1) prioritize tests and (2) help developers categorize and locate the cause of a failure. The outcome of this work will advance the state-of-the-art in test prioritization and localization and provide Ericsson with a more efficient test infrastructure.

Background, Research Approach, and Data

The proposed work intersects two areas of software engineering: test prioritization and empirical software engineering. There are three streams of regression testing research [13]. The first, *minimization*, involves eliminating tests that are redundant or of low value. In the literature, the problem has been reduced to one of code coverage, for example, tests become redundant as the system evolves and more than one test covers the same control flow. As a result, much of the work in this area is algorithmic, such as transforming it into a spanning set problem [6], using divide-and-conquer strategies [1], and greedy algorithms [12]. The second, *selection*, uses the same static analysis techniques such as coverage [11] and slicing [3], but selects tests that cover source files that are at higher risk because they have been changed recently [8]. The third, *prioritization*, orders tests such that expensive, low-value, or long running tests are run after tests that find faults early. We focus on test prioritization because, at Ericsson, once a test fails an engineer must intervene to discover the fault, so an ordering that makes the test run fail early is more cost effective. While early prioritization techniques continued to use coverage measures to gauge priority, more recent approaches incorporate the faults found in past test runs [4] and change relationships among files [9] to identify high value tests.

We combine test prioritization techniques with empirical software engineering. Empirical researchers use historical data to create models to help developers, for example, identify bugs and risky changes [2], locate fault introducing changes [5], and understand the faults found during code review [7]. Relatively little work has combined test archives with information from other artifacts, such as bug reports and code reviews. One of the first studies to look at historical artifacts was done by a colleague at Concordia, Dr. Shihab, during his PhD. In this work, he created a statistical model from the source code history to help developers decide where to write tests for a legacy system at Blackberry [10].

A major factor limiting research into historically based test prioritization is information about past runs and links to other development artifacts. Ericsson collects massive histories of logs, test runs, and development artifacts, in part to provide for reproducibility of software and test lineups and auditing. Much of the current analysis is done manually where developers must sift through this information to locate the root cause of a failure. We plan to automate this process to help in prioritization test and localizing faults.

Research Competence

My research team has extensive training in mining and linking software development artifacts. We have worked extensively with semi-structured datasets that are similar to those at Ericsson and have been able to provide useful statistical models. For example, we have mined millions of code reviews, bug reports, tests results, and StackOverflow discussions. Rigby currently teaches these mining, linking, and analytics

techniques in a graduate course. Rigby has mined test data from Google Chrome in the past. While the Ericsson data will be richer, the techniques remain similar. The novel contribution lies in advancing our knowledge of test prioritization and localization and in improving the efficiency of quality assurance at Ericsson Ottawa.

Research Project and Technology Transfer

The proposal consists of three milestones, which will run for approximately 2 months each. First, we will investigate, clean, and link the available data. Second we will create a test prioritization model (RQ1). Third, we will create a fault localization model (RQ2).

Investigate Process and Data: A masters student will be onsite at Ericsson Ottawa. His goal will be to work with Griffiths's team to understand the quality assurance processes and to clean and link data. Ericssons in-kind contribution will be substantial as they will work with the masters student and help in interpreting the findings. I will visit Ericsson twice a month to help the student. A PhD student, Rahman, who researches release engineering, will also be investigating test prioritization and the impact on release effectiveness and help generalize the masters student findings to other software projects.

RQ1. Test prioritization model: Not all tests are equally important, but all tests must be run. With Ericsson's hard quality requirements, all tests must be run before a change can be released. However, some tests can span days, while others take seconds. Further, some tests have not failed in months, while some are new or fail regularly. Since a failure will stop the build process and require an engineer to intervene, a clear research direction will be to prioritize based on past failures and test running time [4, 14]. A simple ordering that makes low cost, high value tests fail early will make the process more efficient. Adding to these measures, we will develop change risk and social measures, such as risk analysis based on histories of defects across software areas and development teams. The research outcome of this stage will be a better understanding of the archival measures that provide good test prioritization.

RQ2. Fault categorization and localization model. Since fault localization is a wide research area including debugging techniques and statistical bug models, in this proposal, we focus on the narrow question – Did that test fail because of a problem in the development environment or product? This problem is unusually difficult at Ericsson because the environment is exceedingly complex. For example, test equipment is so expensive that developers in Ottawa can request their tests be run on specialized equipment that is only available in Sweden, and vice versa. Once a test fails, a developer must determine the root cause of the failure and whether it is an environment or product problem. Unfortunately, much of the other information in the report is added sporadically. We will investigate the reliability of existing fields in the reports and supplement this with freeform data extracted using existing NLP techniques. Our goal is to train a classifier based on past manual categorizations to help developers differentiate these failure types. In the future, we plan to refine this coarsegrained categorization to a finegrained localization at the file or method level by linking issue reports to other software artifacts.

Benefits to Canada

In 2009, Ericsson acquired the majority of Nortel's North American wireless access business, including its market share and became the world leader in this market. This human expertise remains in Canada where Ericsson employs about 1000 R&D staff in Ottawa. Our contribution will be to make these individuals more productive by improving the efficiency of the test process. If this NSERC Engage collaboration proves to be beneficial, we will extend the project through MITACS. We hope that this contribution leads to a long term relationship with Ericsson Ottawa where we can attack their practical problems and abstract them to the general advancement of knowledge in software engineering.

References

[1] T. Y. Chen and M. F. Lau. Dividing strategies for the optimization of a test suite. *Information Processing Letters*, 60(3):135 – 141, 1996.

[2] M. D'Ambros, M. Lanza, and R. Robbes. An extensive comparison of bug prediction approaches. In *Mining Software Repositories (MSR), 2010 7th IEEE Working Conference on*, pages 31–41, May 2010.

[3] D. Jeffrey and N. Gupta. Test case prioritization using relevant slices. In *Proceedings of the 30th Annual International Computer Software and Applications Conference - Volume 01*, COMPSAC '06, pages 411–420, Washington, DC, USA, 2006. IEEE Computer Society.

[4] J.-M. Kim and A. Porter. A history-based test prioritization technique for regression testing in resource constrained environments. In *Proceedings of the 24th International Conference on Software Engineering*, ICSE '02, pages 119–129, New York, NY, USA, 2002. ACM.

[5] S. Kim, T. Zimmermann, K. Pan, and E. J. J. Whitehead. Automatic identification of bug-introducing changes. In *Proceedings of the 21st IEEE/ACM International Conference on Automated Software Engineering*, ASE '06, pages 81–90, Washington, DC, USA, 2006. IEEE Computer Society.

[6] M. Marré and A. Bertolino. Using spanning sets for coverage testing. *IEEE Trans. Softw. Eng.*, 29(11):974–984, Nov. 2003.

[7] P. C. Rigby, D. M. German, L. Cowen, and M.-A. Storey. Peer review on open-source software projects: Parameters, statistical models, and theory. *ACM Trans. Softw. Eng. Methodol.*, 23(4):35:1–35:33, Sept. 2014.

[8] G. Rothermel and M. J. Harrold. A framework for evaluating regression test selection techniques. In *Proceedings of the 16th International Conference on Software Engineering*, ICSE '94, pages 201–210, Los Alamitos, CA, USA, 1994. IEEE Computer Society Press.

[9] M. Sherriff, M. Lake, and L. Williams. Prioritization of regression tests using singular value decomposition with empirical change records. In *Proceedings of the The 18th IEEE International Symposium on Software Reliability*, ISSRE '07, pages 81–90, Washington, DC, USA, 2007. IEEE Computer Society.

[10] E. Shihab, Z. M. Jiang, B. Adams, A. E. Hassan, and R. Bowerman. Prioritizing the creation of unit tests in legacy software systems. *Software: Practice and Experience*, 41(10):1027–1048, 2011.

[11] A.-B. Taha, S. Thebaut, and S.-S. Liu. An approach to software fault localization and revalidation based on incremental data flow analysis. In *Computer Software and Applications Conference, 1989. COMPSAC 89., Proceedings of the 13th Annual International*, pages 527–534, Sep 1989.

[12] S. Tallam and N. Gupta. A concept analysis inspired greedy algorithm for test suite minimization. *SIGSOFT Softw. Eng. Notes*, 31(1):35–42, Sept. 2005.

[13] S. Yoo and M. Harman. Regression testing minimization, selection and prioritization: a survey. *Software Testing, Verification and Reliability*, 22(2):67–120, 2012.

[14] Y. Yu, J. A. Jones, and M. J. Harrold. An empirical study of the effects of test-suite reduction on fault localization. In *Proceedings of the 30th International Conference on Software Engineering*, ICSE '08, pages 201–210, New York, NY, USA, 2008. ACM.