Application Scenarios for a Customizable System Dynamics Simulation Model of Generic Software Development Processes

Keyvan Khosrovian[1], Dietmar Pfahl[1, 2, 3], Vahid Garousi[1]

[1]Schulich School of Engineering, University of Calgary, Canada
[2]Simula Research Laboratory, Norway,
[3]Department of Informatics, University of Oslo, Norway
{kkhosrov, dpfahl, vgarousi}@ucalgary.ca

Abstract. GENSIM 2.0 is a customizable system dynamics simulation model of generic software development processes which takes as input specifics of software development projects (e.g. size of the code document, headcount of the developer team and their skills) and generates as output a broad range of distinct variables (e.g. software product quality, total project effort) of interest to different users of the model. This technical report is dedicated to illustrate example scenarios of the application of GENSIM 2.0 to address software development process issues. Firstly, it is shown how GENSIM 2.0 could be used to find the most suitable combination of verification and validation techniques in order to achieve defined time, quality and cost goals in a given context. Secondly, it is used to figure out the most promising investments in a development project's workforce. These scenarios represent only a small subset of the numerous situations that GENSIM 2.0 can be applied to.

Keywords. System Dynamics, Customizable Simulation Model, Software Development Process.

TABLE OF CONTENTS

1 Introduction

Managerial aspects of software development projects including planning, resource allocation, workforce training, etc. have a significant impact on their performance measures, i.e., duration, effort and the quality of the final product. However these areas have not received enough attention by the software engineering research community and still most problems involving these issues are often dealt with relying on expert knowledge and intuition only.

Empirical research is essential for developing theories of software development, transforming the art of software development into an engineering discipline and hence overcoming the aforementioned problem. However, empirical theories and knowledge require evidences for efficiency and effectiveness of the tools, techniques or practices in multiple different application contexts.

Controlled experiments and surveys are methods commonly used for empirical research, however, they are costly. Hence, support for making decisions on which experiments and case studies are more worthwhile to spend effort and time on would be helpful. GENSIM 2.0 [1] is a customizable and reusable software development process simulation model developed to address this issue. Inspired by the idea of frameworks in software systems, GENSIM 2.0 consists of a small set of generic reusable components which can ideally be plugged together to model a wide range of different software development processes.

These components capture key attributes of the entities involved in different building blocks of the development processes that affect the project performance measures. What makes the model results interesting and hard to precisely predict, are the numerous complex relationships and influences between each pair of these attributes. Current implementation of GENSIM 2.0 simulates the well-known V-Model software development process. It consists of three development phases (requirements specification, design and code), each consisting of a document development activity and a verification activity (e.g. Inspection) carried out on the developed artifacts, and three validation activities (unit, integration and system test).

GENSIM 2.0 can assist software development process management in many different ways. Following is a list of a small subset of them:

- Evaluating the overall effectiveness and efficiency of different combinations of development, verification, and validation techniques
- Analyzing the overall impact of changes to the workforce characteristics on project performance.

The rest of this report is dedicated to demonstrating usefulness of GENSIM 2.0 using detailed application scenarios related to the above list.

2 Related Work

The idea of using software process simulators for predicting project performance or evaluating the impact of process changes on project performance is not new. Beginning with pioneers like Abdel-Hamid [2], Bandinelli [3], Gruhn [4], Kellner [5], Scacchi [6], and many others, dozens of process simulation models have been developed and applied for various purposes.

[7] and [8] are examples of the application of simulation models to tackle project management problems. In [7], Padberg focuses on scheduling issues and computes optimal scheduling strategies for a set of sample software projects using a stochastic model. In [8], Lee targets at multi-project management and an integration of a System Dynamics model with a multi-project network analysis method, called Critical

Chain Project Management. Lee also proposes a model to identify the restraining factors in various possible scenarios in a multi-project setting.

[9] and [10] are examples of simulation modeling for risk management. In [9], Houston describes an approach to modeling risk factors and simulating their effects as a means of supporting certain software development risk management activities. The effects of six common and significant software development risk factors are studied. In [10] a five step simulation-based method to risk assessment, ProSim/RA, which combines software process simulation with stochastic simulation, is presented.

[11], [12], [13], [14] and [15] are examples of simulation modeling applications for software quality assurance. In [11], the authors propose a procedure to investigate whether increasing test coverage has a genuine additional impact on defect coverage when compared to the impact of increasing test effort. A precise simulation and analysis procedure to analyze the cost-effectiveness of statechart-based testing techniques is presented in [12]. Using the proposed procedure, the cost and fault detection effectiveness of adequate test sets for the most referenced coverage criteria for statecharts on three different representative case studies is investigated. In [13] a dynamic simulation model of an inspection-based software lifecycle process has been developed to support quantitative process evaluation. The model serves to examine the effects of inspection practices on cost, scheduling and quality throughout the lifecycle. In [14] the authors describe the use of a process simulator to support software project planning and management. The proposed modeling approach focuses on software reliability and it is argued that it is just as applicable to other software quality factors, as well as to cost and schedule factors.

Nevertheless, application of simulation modeling in software engineering is not limited to project management, risk management and quality assurance. Over the years, simulation modeling has been applied to tackle a wide range of problems in many different areas. Process engineering, strategic planning, product and requirements engineering, software maintenance and evolution, global software development, software acquisition management and COTS, product lines and training and education are among these areas. For an overview of software process simulation works done in the past 15 to 20 years refer to [16].

In [15] Raffo et al discuss modeling a NASA project using the IEEE 12207 software development process with multiple possible IV&V (Independent Verification and Validation) configurations. The purpose of the research is to quantitatively assess the financial benefits of applying IV&V techniques in software development projects and figuring out the optimal alternatives regarding those benefits. They raise a series of questions that the developed model could be used to answer. Evaluating the costs and benefits associated with implementing a given IV&V technique on a selected software project, evaluating how and to what extent employing a particular combination of IV&V techniques affect the development phase of a project and assessing the impact of additional staff for IV&V on the cost and schedule of the project are examples of these questions. At the end, application of the model to evaluate three different situations is discussed in more detail. Firstly they apply the model to assess the impact of IV&V at different points in the development process on the overall project performance measures. Secondly, the impact of inserting additional IV&V techniques is evaluated and at last the model is used to assess the impact of adding staff. While being good examples of situations that simulation models are helpful, the results presented in [15] cannot be fully analyzed because of the confidentiality of the details of the internal structure of the model.

Inspired by the questions analyzed in [15], in this report, GENSIM 2.0 is applied to answer similar questions. Due to additional possibilities that GENSIM 2.0 offers, it is also used to investigate the effect of workforce skill levels on project performance measures.

It should be noted that GENSIM 2.0, unlike many other previous process simulation models of its kind, has not been custom built to target a specific issue only. Rather it is intended to be reused, customized and applied to tackle emerging software development related problems of any kind.

3 GENSIM 2.0 APPLICATION SCENARIOS

In this section, two different types of problems that GENSIM 2.0 can help to solve are discussed and the results of its application are presented. Initially, in sections 3.1 and 3.2, GENSIM 2.0 is used to find out the most effective and efficient combination of verification and validation techniques with regards to specific time, effort and quality goals. Secondly, in sections 3.3 and 3.4, it is used to find the most promising areas of investment in a project's workforce.

3.1 SCENARIO 1: CHOOSING THE BEST COMBINATION OF V&V TECHNIQUES WITH RESPECT TO PROJECT PERFORMANCE GOALS

One of the important features of GENSIM 2.0 is that its process structure can easily be modified. This scenario exploits this feature of GENSIM 2.0 to address the issue of finding the most suitable combination of verification and validation (V&V) activities considering time, effort and quality goals. It shows the impact of different combinations of V&V activities on project duration, product quality, and effort and how the model could assist decision makers in choosing with the best alternative. Verification activities include Requirements Inspections (RI), Design Inspections (DI) and Code Inspections (CI). Validation activities include Unit Test (UT), Integration Test (IT), and System Test (ST). For each V&V activity there is exactly one technique with given efficiency (i.e., V&V rate) and effectiveness available. A V&V technique is either applied to all of the documents of the related type (e.g., requirements, design, and code documents) or it is not applied at all.

To clearly show how different calibration values can affect result analysis and how the model can assist decision-making in different development contexts, besides the original calibration of the model (Calibration B) this scenario was run using another calibration (Calibration A). The difference between these calibrations is shown in Table 1. In Calibration B, rework effort per detected defects is greater than in Calibration A for defects detected during integration and system testing.

Table 1: Difference between Calibration A and Calibration B

Calibration Parameter	Value	
	Calibration A	Calibration B
Code rework effort for code faults detected in IT	0.6775 PD/Def. [17]	1.0812 PD/Def. [[17], [18]]
Code rework effort for code faults detected in ST	1.0462 PD/Def. [17]	5.6225 PD/Def. [[17], [18]]

PD = Person-Day, Def. = Defect

Generating all the different V&V combinations and their respective simulation results was done automatically using the 'Sensitivity Analysis' feature provided in Vensim®. Figure 1 shows the simulation results for both calibrations of the model (Calibration A above / Calibration B below). Squares represent (Quality, Duration) result value pairs, where quality is measured as the total number of undetected code faults. Triangles represent non-dominated (Quality, Duration) result values, i.e., simulation results to which no other simulation exists with both less undetected defects and less duration.

Obviously, for both calibrations, there exists a trade-off between Quality and Duration. Only looking at the non-dominated solutions, one can see that in order to achieve less undetected defects, more time is needed. In effect, if the goal was to see which combinations of V&V activities should be applied to achieve the target duration, in the case that there are several eligible V&V combinations, a decision-maker could pick the non-dominated solution with the lowest number of undetected defects that is just within the project deadline.

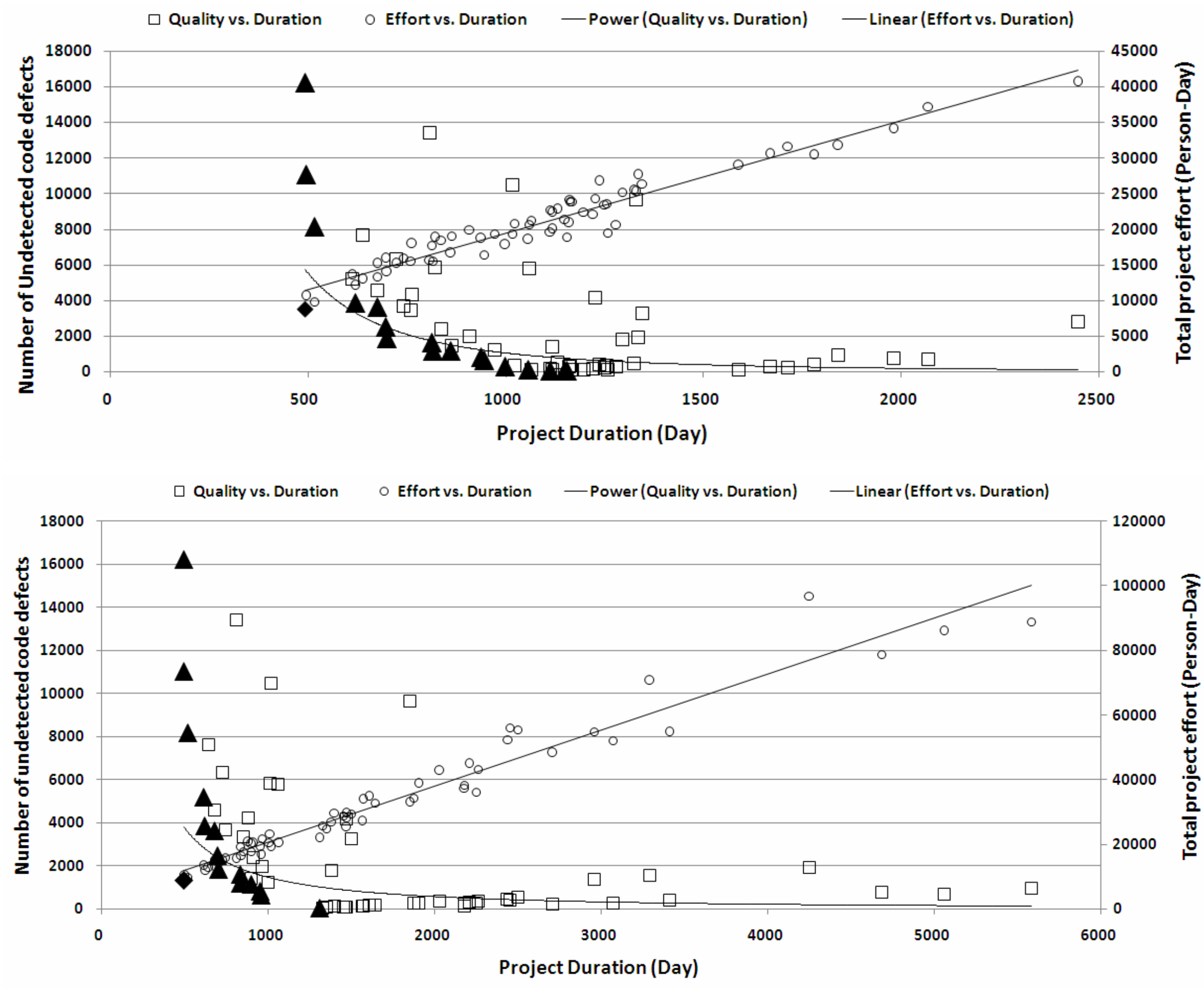

Figure 1: Quality vs. Duration and Effort vs. Duration (Scenario 1 – Calibrations A and B)

Circles represent (Effort, Duration) result value pairs, where effort is measured as the total number of person-days spent on all activities (including rework). The only non-dominated solution is represented by the diamond symbol near the lower left end of the (Effort, Duration)-regression line. The fact that there is no trade-off between effort and duration can be explained by the fact that all simulations use the same workforce.

There is, however, a difference between Calibration A and B. On average, simulations with Calibration B take longer and consume more effort. This can be explained by the fact that Calibration B assumes greater per defect rework effort for code defects found in IT and ST than Calibration A.

Furthermore, a detailed analysis of all $2^6 = 64$ simulations using Table 2 which contains detailed results of the simulation runs per calibration reveals that the set of non-dominated solutions and their rankings differ for cases that involve IT and ST. For example, it turns out that with Calibration B, combination (RI, DI, CI, UT, -, ST) is better than combination (-, DI, -, -, IT, ST) with regards to duration, effort and quality. With Calibration A, combination (RI, DI, CI, UT, -, ST) is better than combination (-, DI, -, -, IT, ST) only with regards to effort and quality, but not with regards to project duration.

Table 2: Simulation results for the Scenario 1

RI	DI	CI	UT	IT	ST	Duration [Day]		Effort [PD]		Quality [UD]	
						Cal. A	Cal. B	Cal. A	Cal. B	Cal. A	Cal. B
0	0	0	0	0	0	449	449	8640	8640	28508	28508
1	0	0	0	0	0	492	492	8836	8836	16241	16241
0	1	0	0	0	0	496	496	10639	10639	11054	11054
1	1	0	0	0	0	517	517	9825	9825	8171	8171
0	0	1	0	0	0	809	809	15701	15701	13422	13422
1	0	1	0	0	0	639	639	12986	12986	7656	7656
0	1	1	0	0	0	611	611	13602	13602	5218	5218
1	1	1	0	0	0	620	620	12085	12085	3850	3850
0	0	0	1	0	0	1017	1017	19370	19370	10498	10498
1	0	0	1	0	0	723	723	15193	15193	6343	6343
0	1	0	1	0	0	675	675	15195	15195	4597	4597
1	1	0	1	0	0	676	676	13342	13342	3633	3633
0	0	1	1	0	0	1060	1060	20655	20655	5775	5775
1	0	1	1	0	0	742	742	15952	15952	3663	3663
0	1	1	1	0	0	697	697	15908	15908	2499	2499
1	1	1	1	0	0	701	701	13989	13989	1845	1845
0	0	0	0	1	0	1329	1850	25282	33224	9641	9638
1	0	0	0	1	0	821	1008	18865	23339	5863	5840
0	1	0	0	1	0	764	877	17946	20936	4307	4242
1	1	0	0	1	0	761	850	15577	17765	3452	3369
0	0	1	0	1	0	1226	1469	24361	28104	4163	4163
1	0	1	0	1	0	838	906	18490	20622	2376	2376
0	1	1	0	1	0	813	831	17792	19241	1621	1620
1	1	1	0	1	0	815	834	15532	16595	1197	1196
0	0	0	1	1	0	1346	1497	26432	29359	3277	3277
1	0	0	1	1	0	910	961	19984	21746	1993	1988
0	1	0	1	1	0	864	894	19050	20316	1455	1447
1	1	0	1	1	0	862	896	16666	17670	1154	1147
0	0	1	1	1	0	1295	1377	25261	26869	1791	1791
1	0	1	1	1	0	972	1000	19390	20490	1202	1224
0	1	1	1	1	0	939	951	18707	19457	824	832
1	1	1	1	1	0	946	958	16448	17005	626	629
0	0	0	0	0	1	2447	10400	40664	162002	2794	2793
1	0	0	0	0	1	1335	4246	27663	96784	1937	1934
0	1	0	0	0	1	1162	3288	23928	70974	1576	1572
1	1	0	0	0	1	1117	2956	20011	54788	1376	1370
0	0	1	0	0	1	1842	5581	31737	88857	942	942
1	0	1	0	0	1	1131	2498	22820	55405	539	539
0	1	1	0	0	1	1022	2026	20835	43041	368	368
1	1	1	0	0	1	1000	1873	17846	34232	272	272

0	0	0	1	0	1	1980	4682	34083	78759	760	760
1	0	0	1	0	1	1324	2435	25434	52430	469	469
0	1	0	1	0	1	1255	2259	23573	43136	347	347
1	1	0	1	0	1	1280	2250	20681	36140	280	279
0	0	1	1	0	1	1780	3408	30441	55019	405	405
1	0	1	1	0	1	1251	1904	23454	39044	257	257
0	1	1	1	0	1	1222	1641	22169	32806	176	176
1	1	1	1	0	1	1258	1564	19545	27398	130	130
0	0	0	0	1	1	2069	5057	37251	86209	697	697
1	0	0	0	1	1	1236	2450	26768	56075	433	431
0	1	0	0	1	1	1164	2205	24184	45164	324	319
1	1	0	0	1	1	1161	2174	20891	37339	264	258
0	0	1	0	1	1	1668	3071	30602	52063	292	292
1	0	1	0	1	1	1112	1604	22778	35053	167	167
0	1	1	0	1	1	1064	1395	21289	29630	114	114
1	1	1	0	1	1	1058	1349	18576	24735	84	84
0	0	0	1	1	1	1714	2704	31714	48584	232	232
1	0	0	1	1	1	1169	1571	23883	34080	142	142
0	1	0	1	1	1	1117	1468	22374	29790	104	104
1	1	0	1	1	1	1114	1465	19666	25545	83	83
0	0	1	1	1	1	1590	2177	28979	38223	126	126
1	0	1	1	1	1	1198	1449	22483	28818	84	86
0	1	1	1	1	1	1151	1326	21397	25697	58	58
1	1	1	1	1	1	1154	1308	18924	22160	44	44

PD = Person-Day, UD = # of undetected defects in the code document

3.2 Scenario 2: Choosing the best combination of verification techniques with respect to project performance goals

This scenario uses only Calibration B as explained above. It shows the impact of different combinations of verification activities and techniques on project duration, product quality, and effort. This scenario assumes that all validation activities UT, IT, and ST are always performed, while verification activities (RI, DI and CI) can be performed or not. If a verification activity is performed, one of alternative techniques A or B can be applied. Compared to A-type verification techniques, B-type techniques are always 10% more effective (i.e., find 10% more of all defects contained in the related artifact) and 25% less efficient (i.e., 25% less size of the related document can be verified per person-day).

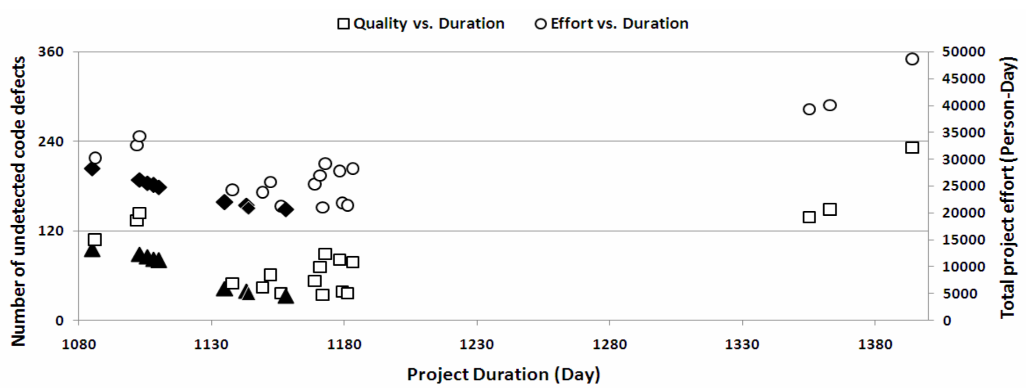

Figure 2: Quality vs. Duration and Effort vs. Duration (Scenario 2 – Calibration B)

The simulation of all possible combinations generates $3^3 = 27$ different results (cf. Figure 2 and Table 1). Similar to what is shown in Figure 1, in Figure 2 the non-dominated solutions are marked by diamonds and triangles. The main difference to Scenario 1 is that in addition to the (Quality, Duration) trade-off there is a simultaneous (Effort, Duration) trade-off. For example, when having a closer look at Table 3 one notices that strictly using B-type techniques in all performed verification activities will always result in less effort consumption and better quality than strictly using A-type techniques. With regards to duration, however, the picture is not so clear. While simulation results using patterns (B, 0, 0), (0, B, 0), (0, 0, B), (0, B, B), and (B, 0, B) indicate shorter duration than corresponding patterns using strictly A-type techniques, simulation results using patterns (B, B, 0) and (B, B, B) show longer durations than corresponding patterns (A, A, 0) and (A, A, A). When there is a mix of A-type and B-type verification techniques, the picture is even more complex and no general conclusions can be made, hence, the results have to be looked at in detail for each individual case.

Table 3: Simulation results for Scenario 2

Case	RI	DI	CI	RI-tech	DI-tech	CI-tech	Duration [Day]	Effort [PD]	Quality [UD]
1	0	1	0		B		1085	28401	96
2	0	1	0		A		1086	30272	108
3	1	0	0	B			1102	32830	135
4	1	0	0	A			1103	34448	145
5	1	1	0	A	A		1103	26216	89
6	1	1	0	B	A		1106	25637	86
7	1	1	0	A	B		1108	25220	82
8	1	1	0	B	B		1110	24892	81
9	1	1	1	A	A	A	1135	22032	43
10	0	1	1		B	A	1138	24297	50
11	1	1	1	B	A	A	1143	21516	40
12	1	1	1	B	B	A	1144	20926	36
13	0	1	1		B	B	1149	23888	45
14	0	1	1		A	A	1152	25916	61
15	1	1	1	A	B	A	1156	21253	38
16	1	1	1	B	B	B	1158	20728	33
17	0	1	1		A	B	1169	25412	54
18	1	0	1	B		B	1171	27094	72
19	1	1	1	A	B	B	1172	21153	35
20	1	0	1	A		A	1173	29187	90
21	1	0	1	B		A	1178	27836	82
22	1	1	1	A	A	B	1179	21865	40
23	1	1	1	B	A	B	1181	21434	38
24	1	0	1	A		B	1183	28302	79
25	0	0	1			B	1355	39446	139
26	0	0	1			A	1363	40287	149
27	0	0	0				1394	48683	233

PD = Person-Day, UD = # of undetected defects in the code document

3.3 Scenario 3: Analyzing the effect of workforce headcount on project performance

The headcount of the workforce available for a project and their capabilities in carrying out different activities in the project have a significant impact on the project's performance. GENSIM 2.0 enables the project management to analyze this impact, taking into account all the mutual influences between the characteristics of the staffing profile, the sequence of activities, organizational policies for workforce allocation and other factors involved in the overall development process. The scenario presented in this section shows an example of the situations that GENSIM 2.0 could assist the management by providing estimates of the potential effects of the changes to a project's staffing profile.

To achieve increased reusability, in the implementation of GENSIM 2.0, organization-specific policies are extracted from the SD model and incorporated into external Dynamic Link Libraries (DLL) which allows for easy modification of these heuristics and algorithms [1]. The workforce allocation is an example of such an algorithm. The current workforce allocation algorithm in GENSIM 2.0 which is also used for the purpose of the scenarios represented in this section is explained in detail in [1].

Characteristics of the available workforce in GENSIM 2.0 are represented by an $n \times m$ matrix S, as shown in equation 1. In this matrix n is the headcount of the available workforce, m is the number of activities which are carried out in the development life-cycle, and s_{ij} represents the skill level of the i^{th} employee in carrying out the j^{th} activity. Skill level of 1 means that the employee is fully skilled in carrying out the task and skill level of 0 means that he/she is not able to carry out the task. In all the runs of this scenario, since we are only concerned with the number of employees that can carry out the activities, it is always assumed that all of them are fully skilled.

$$S_{n\times m} = \begin{bmatrix} s_{11} & \cdots & s_{1m} \\ \vdots & \ddots & \vdots \\ s_{n1} & \cdots & s_{nm} \end{bmatrix}, s_{ij} \in [0,1]$$

Equation 1: Staffing profile representation in GENSIM 2.0

In this scenario, GENSIM 2.0 is used to evaluate the effect of doubling the headcount of a project's available workforce on its performance measures, i.e., effort, quality and duration. The analysis is performed on two extreme cases. In the first case, each activity is carried out by only one developer. In the second case, all activities can be carried out by all available employees. Any other case between these extremes could be investigated similarly.

Case 1: In this case, the initial workforce consists of 6 employees and each activity can be carried out by only one of them. Hence, the staffing profile matrix could look like the example given in Equation 2. As can be seen, in this example each employee is capable of carrying out two consecutive activities. The simulation run with this staffing profile is referred to as the baseline run.

$$S_{6\times 12} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Equation 2: Initial Staffing profile matrix for Scenario 3 - Case 1

The effect of doubling the headcount of the workforce is analyzed in two different ways. Firstly, any of the activities can be carried out by only one of the employees and any of the employees can carry out only one activity. Therefore, the staffing profile matrix is defined as shown in Equation 3. The simulation run with this staffing profile is referred to as run A. Secondly, any of the activities can be carried out by two of the employees but each of these two can carry out two activities. Hence, the staffing profile matrix is specified as shown in Equation 4. The simulation run with this staffing profile is referred to as run B.

$$S_{12\times12}=\begin{bmatrix}1&0&0&0&0&0&0&0&0&0&0&0\\0&1&0&0&0&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0&0&0&0\\0&0&0&0&1&0&0&0&0&0&0&0\\0&0&0&0&0&1&0&0&0&0&0&0\\0&0&0&0&0&0&1&0&0&0&0&0\\0&0&0&0&0&0&0&1&0&0&0&0\\0&0&0&0&0&0&0&0&1&0&0&0\\0&0&0&0&0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&0&0&0&0&1\end{bmatrix}$$

Equation 3: Staffing profile matrix for run A of Scenario 3 - Case 1

$$S_{12\times12}=\begin{bmatrix}1&1&0&0&0&0&0&0&0&0&0&0\\1&1&0&0&0&0&0&0&0&0&0&0\\0&0&1&1&0&0&0&0&0&0&0&0\\0&0&1&1&0&0&0&0&0&0&0&0\\0&0&0&0&1&1&0&0&0&0&0&0\\0&0&0&0&1&1&0&0&0&0&0&0\\0&0&0&0&0&0&1&1&0&0&0&0\\0&0&0&0&0&0&1&1&0&0&0&0\\0&0&0&0&0&0&0&0&1&1&0&0\\0&0&0&0&0&0&0&0&1&1&0&0\\0&0&0&0&0&0&0&0&0&0&1&1\\0&0&0&0&0&0&0&0&0&0&1&1\end{bmatrix}$$

Equation 4: Staffing profile matrix for run B of Scenario 3 - Case 1

Simulation results of run A, run B and the baseline run are shown in Table 5. It can be seen that because in run B each employee can carry out two activities and could be potentially allocated to any of them, run B yields a much greater improvement than run A with regards to the duration of the project. This could be explained by existing constraints inherent to the process structure. For example, requirements specification verification activity can only begin when the requirements specification development activity is finished. Therefore, there is only a small overlap between the periods that each of these activities requires allocated workforce and that is the period when the verification is being carried out and the detected faults are being corrected meanwhile. As a result in run B, in most of the times when there is requirements specification development activity to be done, two developers are assigned to the activity and there is no competition between the development and verification activity.

Quality remains the same in all the runs, because the skill levels of all the employees remain constant across different runs. The difference in the effort estimations is explained by the fact that the time step chosen for the simulation runs is one whole day. So, the employees are re-allocated to the activities on a daily basis. In cases that there is little work left to be done in any of the activities, the model still allocates workforce to that activity for the whole day which in turn causes the resulting effort estimations slightly different from the actual effort that has to be spent for that activity.

Table 4: Simulation results for Scenario 3 - Case 1

Run	Duration [Day]	Difference in Duration from baseline	Effort [PD]	Difference in effort from baseline	Quality [UD]	Difference in quality from baseline
Baseline	1088	0%	901	0%	2	0%
A	1080	-0.73%	1061	+17.75%	2	0%
B	610	-43.93%	1093	+21.30%	2	0%

PD = Person-Day, UD = # of undetected defects in the code document

Case 2: In this case, the initial workforce consists of 6 developers and any activity can be carried out by any of the employees, i.e., each of them is capable of carrying out any of the activities. Hence, the staffing profile matrix is defined as illustrated in Equation 5. The simulation run with this staffing profile is referred to as the baseline run.

$$S_{6\times 12} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Equation 5: Initial Staffing profile matrix for Scenario 3 – Case 2

The doubling effect is analyzed for a team of workforce with 12 developers with the same pattern of capabilities as the baseline run, i.e., each of the employees is capable of carrying out any of the activities. The simulation run with this staffing profile is referred to as run A. The simulation results of the two runs of this case are shown in Table 6.

Table 5: Simulation results for Scenario 3 – Case 2

Run	Duration [Day]	Difference in Duration from baseline	Effort [PD]	Difference in effort from baseline	Quality [UD]	Difference in quality from baseline
Baseline	280	0%	1005	0%	2	0%
A	154	-45%	1025	+2%	2	0%

PD = Person-Day, UD = # of undetected defects in the code document

As shown in Table 6, the estimated duration of the project in run A is reduced by 45% percent. The reason why this effect is not estimated as 50% is that some work can be finished within one day whether 6 or 12 developers are allocated. As a result, since the simulation time step is one day, the duration of that particular one-day work remains equal for both runs. The difference in the effort estimates results from the same reasons as explained in Case 1.

Any case in between the above extreme cases, involving arbitrary settings of the staffing profile matrix, could be investigated in the same manner. For example, with the staffing profile illustrated in Equation 6, duration of the project is estimated to be 232 days while the effort spent on the project is estimated to be 1135 person-days and the number of undetected defects in the code is estimated to be 2 defects.

$$S_{12\times 12} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

Equation 6: Staffing profile matrix with arbitrary settings

3.4 Scenario 4: Analyzing the effect of workforce skill levels on project performance

This scenario shows the application of GENSIM 2.0 to analyze the effects of hiring better skilled workforce or training the current workforce on the project's performance and to figure out which type of skill sets is more worth the investment. GENSIM 2.0 assumes that for any of the employees, a skill level, a real number $s \in [0, 1]$, can be given for any of the activities of the project to specify his/her skill level in carrying out that activity. If providing the skill level with such accuracy is not possible and the employees' skill levels could only be specified on an ordinal scale a mapping from the ordinal scale onto [0,1] could resolve the issue. For example if the employees' skill levels is provided on an ordinal scale with five ordinals including excellent, good, medium, weak, and unable to do, we can map them onto [0,1] using a table as shown in Table 7.

Table 6: Example of mapping skill levels from ordinal to ratio scale

Value on ordinal scale	Value on ratio scale
Unable to do	0
Weak	0.25
Medium	0.5
Good	0.75
Excellent	1

The effect of the employees' skill levels on GENSIM 2.0 parameters is twofold. Whenever the skill level of an employee is increased, the speed with which he/she performs that activity is increased while his/her chances of making a mistake decreases. For example, if the skill level of testers is increased, the speed with which they test the artifacts increases while the effectiveness of the testing technique in defect detection increases. If the skill level of developers increases the speed with which they develop/rework the artifacts increases while the number of defects they inject in the artifact decreases. Because of the lack of reliable data on the magnitude of the effect of workforce skill levels on different parameters, it was assumed that all the parameters affected by the workforce skill levels increase/decrease proportionate to the average skill level of the employees i.e. if the average skill level of the system testers is 0.5, then the effectiveness of the system testing technique will drop from its reported optimal value by 50 percent.

The concrete question that this scenario answers is finding out the effect of hiring better skilled or training developers, verifiers and testers on project duration and effort and the quality of the final product. The question is analyzed for two example cases.

Case 1: In this case, the staffing profile matrix is specified as the run A of Case 2 of section 3.3, i.e., 12 employees that each of them could potentially carry out any of the activities. The scenario includes four different simulation runs with differences in their inputs as illustrated in Table 7.

Table 7: Differences in inputs of the runs of Scenario 4 – Case 1

Run	Average skill level of developers	Average skill level of verifiers	Average skill level of testers
Baseline	0.5	0.5	0.5
A	0.75	0.5	0.5
B	0.5	0.75	0.5
C	0.5	0.5	0.75

Different output variables of the model could be used to analyze different effects. Table 8 includes the most important simulation results corresponding to the project's performance measures.

Table 8: Simulation results for the four different runs of the first case

Run	Duration [Day]	Difference in Duration from baseline	Effort [PD]	Difference in effort from baseline	Quality [UD]	Difference in quality from baseline
Baseline	956	0%	4796	0%	502	0%
A	786	-17.78%	3924	-18.18%	418	-16.73%
B	500	-47.69%	2879	-39.97%	230	-52.3%
C	917	-4.07%	4645	-3.14%	158	-68.52%

PD = Person-Day, UD = # of undetected defects in the code document

As shown in Table 9, if the main concern of the project management is the quality of the final product, improving the average skill level of the testers would result in more improvement compared to improving the average skill level of the verifiers or the developers. However if the effort or the duration of the project is considered as well, the third case yields the smallest improvement with regards to these factors. Thus, in order to decide on the group of workforce that is going to be invested in, priorities of the project management have to be taken into account and trade-offs have to be analyzed. It is also worth to pint out why the third case shows more improvement in quality than the second case. This is due to the fact that the model assumes that testing techniques detect a certain percentage of the defects within the code regardless of the total number of defects in the code. The effect of the total number of defects in an artifact on the effectiveness of a related verification and validation technique was not considered in the model because no sufficient data was available.

Case 2: In this case, the workforce consists of 70 developers and each developer can potentially carry out only one activity. The number of developers that can carry out each of the activities is shown in Table 10. Besides its ability to generate estimates of the global effects of changes in a project's staffing profile, this scenario demonstrates that GENSIM 2.0 can easily handle staffing profiles of large development projects. Any of the simulation runs of this case take approximately 15 seconds which is very close to the elapsed time of the runs with the small cases.

Table 9: Workforce information for Scenario 4 – Case 2

Activity	Number of developers
Requirements specification development	6
Requirements specification verification	1
Design development	12
Design verification	3
Code development	23
Code verification	5
Unit test case development	5
Unit test	5
Integration test case development	5
Integration test	5
System test case development	10
System test	10

Four different simulation runs with differences in their input similar to those of Case 1 are analyzed. The results of the simulation runs are shown in Table 10.

Table 10: Simulation results for Scenario 4 – Case 2

Run	Duration [Day]	Difference in Duration from baseline	Effort [PD]	Difference in effort from baseline	Quality [UD]	Difference in quality from baseline
Baseline	545	0%	4872	0%	502	0%
A	446	-18.17%	3986	-18.19%	418	-16.67%
B	329	-39.63%	2984	-38.74%	230	-54.12%
C	524	-3.85%	4729	-2.92%	158	-68.60%

PD = Person-Day, UD = # of undetected defects in the code document

It can be seen that, similar to the first case, if the major concern is quality of the final product, investing in the training the testers is the best choice. However, if duration and effort are important as well, investing in verifiers is the best alternative.

4 Conclusions and Future Work

This technical report shows four sample application scenarios for GENSIM 2.0. These sample applications relate to two of the major concerns of project managements. Firstly, choosing the best combination of V&V techniques and secondly choosing the best staffing profile for a project with regards to the project performance goals. It should be noted that this is only a small subset of the vast range of problems that GENSIM 2.0 could assist the project management in solving. Unlike many other simulation models that target specific and focused software development process issues, GENSIM 2.0 is intended to be reused and customized to assist in tackling potentially any kind of software development process issue.

From the results presented in this technical report, it can be concluded that generally investing more in the verification activities and stopping defects from propagating to downstream phases always results in better quality of the final product while considerably reducing the overall project's duration and effort. This conclusion has been stated in the software engineering literature many times before. However, GENSIM 2.0 facilitates analysis of specific projects in specific contexts with specific constraints and goals.

Future work regarding application scenarios of GENSIM 2.0 will involve re-calibrating, reusing, customizing and applying it to deal with different kinds of software development process issues in real-world industrial development environments.

References

[1] K. Khosrovian, D. Pfahl, and V. Garousi, "GENSIM 2.0: A Customizable Process Simulation Model for Software Process Evaluation," Schulich School of Engineering, University of Calgary SERG-2007-07 and Simula TR 2008-01, 2007.

[2] T. Abdel-Hamid and S. E. Madnick, *Software project dynamics: an integrated approach*: Prentice-Hall, Inc., 1991.

[3] S. Bandinelli, A. Fuggetta, L. Lavazza, M. Loi, and G. P. Picco, "Modeling and improving an industrial software process," *Software Engineering, IEEE Transactions on,* vol. 21, pp. 440-454, 1995.

[4] G. Volker and S. Armin, "Software Process Validation Based on FUNSOFT Nets," in *Proceedings of the Second European Workshop on Software Process Technology*: Springer-Verlag, 1992.

[5] M. I. Kellner and G. A. Hansen, "Software process modeling: a case study," in *System Sciences, 1989. Vol.II: Software Track, Proceedings of the Twenty-Second Annual Hawaii International Conference on*, 1989, pp. 175-188 vol.2.

[6] P. Mi and W. Scacchi, "A knowledge-based environment for modeling and simulating software engineering processes," *IEEE Transactions on Knowledge and Data Engineering,* vol. 2, pp. 283-289, 1990.

[7] F. Padberg, "A study on optimal scheduling for software projects," *Software Process: Improvement and Practice,* vol. 11, pp. 77-91, 2006.

[8] L. Bengee and M. James, "Multi-Project Management in Software Engineering Using Simulation Modelling," *Software Quality Control,* vol. 12, pp. 59-82, 2004.

[9] D. X. Houston, G. T. Mackulak, and J. S. Collofello, "Stochastic simulation of risk factor potential effects for software development risk management," *Journal of Systems and Software,* vol. 59, pp. 247-257, 15 December 2001.

[10] D. Pfahl, "ProSim/RA — Software Process Simulation in Support of Risk Assessment," in *Value-Based Software Engineering*, 2006, pp. 263-286.

[11] L. Briand and D. Pfahl, "Using Simulation for Assessing the Real Impact of Test-Coverage on Defect-Coverage", *IEEE Transactions on Reliability*, vol. 49, no. 1, pp. 60-70, 2000.

[12] L. C. Briand, Y. Labiche, and Y. Wang, "Using simulation to empirically investigate test coverage criteria based on statechart," in *Software Engineering, 2004. ICSE 2004. Proceedings. 26th International Conference on*, 2004, pp. 86-95.

[13] R. J. Madachy, "System dynamics modeling of an inspection-based process," in *Software Engineering, 1996., Proceedings of the 18th International Conference on*, 1996, pp. 376-386.

[14] I. Rus, J. Collofello, and P. Lakey, "Software process simulation for reliability management," *journal of Systems and Software,* vol. 46, pp. 173-182, 15 April 1999.

[15] D. M. Raffo, U. Nayak, S. Setamanit, P. Sullivan, and W. Wakeland, "Using software process simulation to assess the impact of IV&V activities," *IEE Seminar Digests,* vol. 2004, pp. 197-205, 2004.

[16] M. Muller and D. Pfahl, "Simulation Methods," in *Guide to Advanced Empirical Software Engineering*: Springer London, 2008, pp. 117-152.

[17] W. Stefan, "A literature survey of the quality economics of defect-detection techniques," in *Proceedings of the 2006 ACM/IEEE international symposium on Empirical software engineering* Rio de Janeiro, Brazil: ACM, 2006.

[18] L. L. C. W. Lars-Ola Damm, "Faults-slip-through - a concept for measuring the efficiency of the test process," *Software Process: Improvement and Practice,* vol. 11, pp. 47-59, 2006.