Evaluating The Effectiveness of Global eXtreme Programming Framework through its Artifacts

Ridi Ferdiana[1], Lukito Edi Nugroho[1], Paulus Insap Santoso[1], Ahmad Ashari[2]

[1]Department of Electrical Engineering and Information Technology, Gadjah Mada University
[2]Department of Computer Science and Electronics, Gadjah Mada University
Yogyakarta - Indonesia
ridi@te.gadjahmada.edu, lukito@mti.ugm.ac.id, insap@te.gadjahmada.edu, ashari@ugm.ac.id

Abstract: Every framework needs an evaluation. The evaluation determines the capability of the framework to support the phases, activity, roles, and product of the software engineering lifecycle. It is also to make sure that the proposed framework can be adopted in real project and performed better than its predecessor. However, the standard of framework evaluation is somewhat limited. Development team chooses the framework based on the previous real project experience and others case studies. This paper will introduce a novel approach to evaluate the agile software engineering framework through its artifacts. An example of a reference model developed for the Global eXtreme Programming framework (GXP). GXP is used a case study to illustrate how the approach may be applied.

Keywords: Software engineering Framework, Evaluation, Artifacts, Agile, Global eXtreme Programming, Global Software Development

1. Introduction

Software engineering evaluation usually happens in organization. The term organization is meant to apply to a software development group in a company who wants to build software. There are several software engineering frameworks that exist. Therefore, the organization usually does the evaluation to choose which framework that appropriates for the project.

Evaluations are context-dependent, which mean evaluations result can't become to be the best in all circumstances. It's possible that an evaluation in one organization being identified as superior, but similar evaluation in another organization would come to a different conclusion [2]. For example, suppose two organizations compare the multi-site development using the Global Software Development (GSD) framework [7]. One organization can say that GSD provide better communication values. Therefore, GSD is enough for them. However, other organization might say that GSD provides better in the term's artifacts. Hence, the difference in the results of the evaluation might be due to properties that assessed not the method that used.

This paper specifically proposes the evaluation method for the organization that implement agile for their multi-site software development. The paper chooses GXP framework as a framework that combines agile process, XP method, and multi-site development model [1].

One of the valuable information that stored in the multi-site development is artifacts. Artifacts are documentation that lives in a software development project. In multi-site software development, artifacts hold ultimate sources for the project information because of several reasons such as follows.

1. Rather than source codes that only understand by the developers, artifacts can be learnt by any stakeholders in projects.
2. Artifacts can be updated together by any people who join the project in multi-site development.
3. Artifacts can works as offline reference when the peers in the site team can't work together during the geographical or time zone difference.

Based on those facts this paper will use artifacts as an evaluation object.

The results of the evaluation provide a comparison context. Evaluation context in the research is a comparative evaluation. Therefore, it assumes that there are several alternative ways to do the same thing and identify which of the alternatives is the best in specific circumstances. In addition, comparison will be made against an ideal framework. This paper will compare the GXP framework with GSD Framework.

The rest of paper is organized as follows. First, we discuss the existing solution in the software engineering framework evaluation. Secondly, we describe our research approach to synthesize and do the evaluation. The research then reports the result by a discussion of the implication of those results, limitation of the work and future research directions.

2. Current Research Solution

In order to get a valid evaluation results, this research will follow prevailing evaluation method called DESMET [2]. DESMET method is intended to help an evaluator to plan and execute evaluation exercise, which is concerned with the evaluations methods and tools. There are several evaluation types in DESMET, which are establishing measurable effects of using objects (Quantitative) or establishing object appropriateness (Qualitative). There are also circumstances which the hybrid (Qualitative and Quantitative) method is used for selected objects. DESMET evaluation is organized through three different ways, which are formal experiment, case study, and survey. Each way can be executed through quantitative, qualitative, or hybrid approach. Figure 1 displays the DESMET evaluation process.

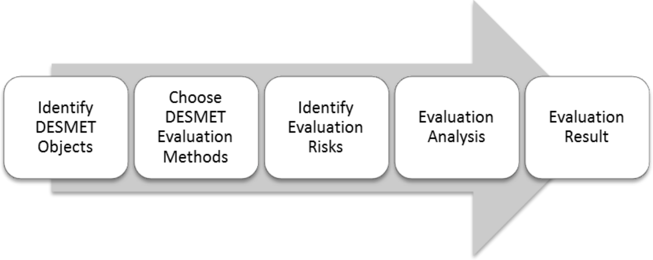


Figure 1. Framework syntax block diagram

2.1 Identify DESMET objects and evaluation method

In the research context, objects are the frameworks that are evaluated. The objects are evaluated through an evaluation method. DESMET provides several evaluation methods. Table 1 provides the evaluation method that can be selected by the evaluator.

Table 1. DESMET Evaluation method [3]

Evaluation method	Conditions favoring method
Quantitative experiments	Benefits clearly. Staff available for taking part in experiment. Method/tool related to single task Benefits directly measurable from task output. Relative small learning time. Desire to make context independent.
Quantitative case studies	Benefits quantifiable on a single project. Benefit quantifiable prior to product refinement. Stable development procedures. Staff with measurement experience. Timescales for evaluation equal with normal projects.
Quantitative surveys	Benefits not quantifiable on a single project. Existing database of project achievements. Projects with experience of using method/tool.
Feature analysis – screening	Large number of methods/tools to assess. Short timescales for evaluation exercise.
Feature analysis case study	Benefits difficult to quantify. Benefits observable on a single project. Stable development procedures. Tool/method user population limited. Timescales for evaluation equal with normal projects.
Feature analysis experiment	Benefits difficult to quantify. Benefits directly observable form task output. Relative small learning time. Tool/method user population very varied.
Feature analysis survey	Benefits difficult to quantify. Tool/method user population very varied. Benefits not observable on single project. Projects with experience of using method/tool.
Qualitative effects analysis	Available of expert opinion assessments of methods/tools. Lack of stable development procedures. Requirement to mix and match method/tool. Interest in evaluation of generic methods (process)/tool.
Benchmarking	Method/tool not human-intensive. Output of method/tool able to be ranked.

Since DESMET provides several ways to evaluate the objects, it needs to identify and select which evaluation method is the most appropriate. DESMET provides specific criteria that use to determine research circumstances, which are:

1. The evaluation context.
2. The nature of the expected impact of using the objects.
3. The nature of the objects to be evaluated.
4. The scope of impacts of the objects.
5. The maturity of the objects.
6. The learning curve associated with the objects.
7. The measurement capability of the organization undertaking the evaluation

The evaluation context of the research is to monitor changes as part of the process improvement program which can involve the evaluation of the proposed framework. The nature of the impact of the research is qualitative impact e.g. better visibility of progress, rapid development and better cost efficiency. The impacts are developed through Qualitative Effects Analysis or Feature Analysis that described in DESMET.

The scope impact of the evaluation has two majors dimension, which are product granularity and extent of impact. Product granularity identifies that the framework is applied to whole development life cycle. The extent of impact of the framework is likely to be felt over the GSD process work flows which are requirement engineering, project planning, architecture design, and product development.

DESMET encourages that the maturity of the framework indicates the extent to which there is likely to be information about it readily available. Since the framework is not used on commercial projects, DESMET states that there would be not sufficient information about the object to warrant a survey (quantitative or qualitative).

Learning curve discusses the time it would take by the organization to learn the objects. Learning time is defined into two aspects, which is time required to understand the principles and time to become proficient in its use. This learning curve related directly with the maturity of the organization. DESMET assumed that the evaluation capability is divided on 4-point ordinal scales, which are:

1. Level 1: Severely limited evaluation capability. The organization does not have well-defined standards.
2. Level 2: Qualitative evaluation capability. The organization has well-defined standards for software development, and adherence to those standards is monitored.
3. Level 3: Quantitative and qualitative evaluation capability. The organization has well-defined standards for software development, and adherence to those standards is monitored.
4. Level 4: Full evaluation capability. The organization has well-defined standards for software development, and adherence to those standards is monitored.

2.2 Identify Evaluation Risks

DESMET implicitly reports several risks that might occur when choosing its evaluation method. Relative risk, cost risk, and human risk are the main risks that happen during its evaluation [4]. DESMET identified two risks elements, which are false positive (The results may imply that a method or tool is beneficial when it is not) and false negative (The results may imply that a method or tool is not beneficial when it is beneficial). The extent of the risk varies for the different methods. The research should ensure that the evaluation method which it has been advised to use will give sufficient confidence that the results of the evaluation are correct. In this setting, "sufficient" must be assessed in relation to the potential cost of an incorrect decision. An incorrect decision would be regarded as serious if a large investment decision was affected, or the framework was to be used to produce safety-critical applications.

Relative risk is the first risk that identified overall risk that exists in selected method. DESMET associated the relative risk into five classes, which are very low, low, medium, high, and very high. Selected method in the research which is featuring analysis case studies is identified as the method with low relative risk. The risk will be lowered when replication and randomization are done with additional experiment. However, the feature analysis case studies identified still subjective by the organization that used the framework.

Cost risk is the risk that should be proportional to the consequential benefits if the framework were judged beneficial and introduced, less the expected investment and introduction costs. As an initial crude guideline, the relative cost of the various evaluations methods are shown as high, medium, low, very low. Feature analysis experiment has high relative cost rather than the others. The cost of experiments is due to a number of different staffs undertaking what may be non-revenue-producing work. The feature analysis experiments have a cost basis like a staff cost and direct cost. Staff cost is cost, which is expanded to define the experiment, devise a scoring method, familiarize the evaluator(s) with the framework, complete the feature analysis questionnaire, collate a result, and produce evaluation reports. Direct costs arise from hardware need, software (including the proposed tools from the framework), and support training for the method.

The attitude and motivation of participants in an evaluation exercise can misrepresent its results. The distortion effects come about because the behavior of staff participating in an evaluation exercise is at adjustment with their working attitude. It can be over-optimistic assessment or pessimistic outcomes. Those behaviors also called as sociological effects [6].

DESMET provides some recommendations that to reduce the risks that happen because of relative risks, cost risks, and human risks which are.

1. Doing the experiment separately from the real project with the separately budget.
2. Successful completion of the experiment project is a formal success of the experiment success.
3. The experiment should be treated as a real project. Therefore, the result will be as real as the real projects.
4. The roles and responsibilities is defined and understood by the people although they are in blind mode.

Those recommendations will be blended through further evaluation analysis that described in the following section.

2.3 Evaluation analysis

In its simplest form, feature analysis provides yes/no response to the existence of a particular property. For analogy example, considers when an organization bought the notebook for their productivity workforce, they might list all properties that you believed to be requirements of the notebook and then allocate a "tick" or "cross" for each property for each notebook candidate, according to whether it obsessed that property. The organization would then count the number of ticks that each candidate had received. Those with the highest counts would offer a short list of candidates on which to carry out a more detailed evaluation to decide their relative value for cash or some other conditions for finally deciding which one to choose.

Implementing the features in the evaluation process will gain some advantages of feature analysis. Simple pre-requisites, can be executed to any required level of detail, and can be applied to any type of method; process and tools are key advantages of feature analysis. In addition, feature analysis also has major advantage that it is not restricted to technical evaluations only, but also managerial and business acquisition evaluations.

Feature analysis also has several disadvantages, which are subjectivity, inconsistencies, collating score, and generating too many features. Feature analysis is been based on judging methods against some "evaluation criteria" which are identified subjectively and context dependent. There is also a problem of inconsistency in scoring between different assessors. If different assessors evaluate different tools, they may have different degrees of familiarity with an understanding of the method/tool. The various score has to be collated and compared to decide the relative order of merit of the methods or tools. Another problem of feature analysis is that hundreds of features may be identified. This makes performing an assessment of a specific method/tool very time consuming and makes it difficult to analyses all the scores. Knowing the disadvantages and keep it in balance will provide sufficient understanding of the assessed framework.

2.4 Evaluation Result

The objective of the evaluation is usually to provide input into a decision about whether to adopt a framework for use by organization. Feature analysis experiment will compare the proposed formalized framework with legacy distributed software development framework. The framework evaluation will normally be intended to carry a specific purpose in qualitative and quantitative degrees. The result of the evaluation needs to provide information in the following areas:

1. Suitability of purposes. It discusses the appropriateness of the framework
2. Economic issue. It discusses the investment and the return of investment when adopting the framework.
3. Drawbacks. It discusses any aspects that make the framework less attractive.

4. Advantages. It discusses any aspects that make the framework more attractive.

In addition, the evaluation method should also help clarify the important features of the method or tool in the context of the organization environment. This can be done easily by identify the framework with the others.

3. Evaluation Methodology

Although DESMET provides a comprehensive guide to evaluate the method and tools, it is only providing a generic approach to do an evaluation. The research has specific need to evaluate the multi-site agile development. Therefore, it needs a modification in the usability of DESMET.

The research chooses case studies approach that already described generic in DESMET. The idea of case study approach is a means of evaluating a framework as part of the normal software development activities undertaken by an organization through real projects. The research discusses the implementation of case studies into three main phases, which are the evaluation preparation, evaluation implementation and evaluation result.

3.1 Evaluation Preparation

Evaluation preparation deals in several steps of feature analysis case studies. They are identifying the case study context, define and validate the hypothesis, select the host projects, identify the method of comparison, minimize the effect of confounding factors, and plan the case study.

Identify the case study context. For evaluations, the research needs both projects to evaluate "which is better" as well as how and why. The how and why aspect can provide valuable awareness into why technology results in better results. The framework which is wanted to evaluate is called treatments. In order to determine whether a treatment is beneficial, it needs to compare it with an alternative treatment or with the currently used framework. The framework that presently uses is mentioned as the control treatment. The control treatment provides a baseline of information to enable comparison to be made. The case study would not only be interested in deciding whether the proposed framework is better than the current framework. It's also needed to define fully to provide the guidance so that the experiment could be replicated.

The research selects two candidate frameworks. The first framework is original GSD framework that used in GSD handbook. Original GSD framework will be used as a controlled project. Controlled project is a real world project which is used GSD process and Unified Process method as a framework to build distributed software development. Both combinations are gently called as legacy GSD. The second framework is GXP framework. GXP framework will be used as an experiment project. Experiment project is a simulation project that has the same people skill set as the controlled project, same technical complexity of the project, and same situation in the environment. The similarity of the project will lead an acceptance for the framework to be evaluated throughout several assumptions of circumstances.

Candidate framework also identified the experiment contexts. The experiment context sets the objective and limitations within which is experimented must operate. The evaluation identifies the experiment sponsor, the available resources, time scales, and the importance of the experiments for the organization. Table 2 provides fact tables for case study context.

Define and validate the hypothesis. This step is restating an evaluation goal in a testable manner. The hypothesis in this research is based on the identification of a particular problem in the distributed software development process which is proposed framework is intended to solve.

Defining the hypothesis is started by specifying the goal of a case study, the framework that wishes to evaluate, the aspect of the framework that is interested in investigating, and an effect in the variable that related directly with the evaluation. The goal of a case study is to identify benefits and weakness of the proposed framework compared with the current framework. GXP framework is a framework that wishes to evaluate, and legacy GSD is a framework that works as the evaluation baseline.

Table 2. Case studies contexts Example

The context of case study planning	Controlled Project	Experiment Project
Case study sponsor	Manufacturing organization	IT Research organization
Organization resources for the case studies	3 person client, 7 development members	2 person client, 5 development members
Case studies timescales	6 months	6 months
Case study importance for the organization	Internal web development for main business process	Project management web application for internal organization

The feature analysis case study uses two main variables, which are responses variables and states variables. Response variables are variables, which is expected change be different pursuant to applying the treatment (i.e. faster than before, efficient than previous, etc.). State variables are factors that characterize the experiments and can influence your evaluation results (i.e. application area, system type, organization model, etc.).

Response variables related directly with case study results. It typically developer productivity and product quality, which are expected to change or to be different pursuant to applying the treatment. The research identifies several response variables based on agile artifacts as follows.

1. Burndown trend pattern. It defines how the project velocity based time and remaining of works. It reflects the productivity of the team.
2. Defect rate. It stores numbers of errors on iteration. It reflects the software quality regarding by the defects.
3. Check-in operation. It stores the historical integrations by the developer to the system. It reflects the continuous improvement of the team.
4. Investment costs. It summarized altogether the development process investments. It reflects the efficiency of investment for the framework regarding the tools, communication, and method investments.
5. Communication pattern. It defines the communication trend between member and client. It counts the

numbers of communications through messages (Email, Instant Messaging) and others online communication.

State variables define the characteristics project that typical project has in the organization. The state variables are described as follows.

1. Application area of the projects. It defines the kind of the project that developed in case study.
2. Framework that used. It defines the software engineering framework that used in case study.
3. Business type. It defines the business type that runs by the organization.
4. Scale of projects. It defines the product size range.
5. Complexity level of problems. It defines the complexity of the software regarding of the business process domain.
6. Quality and experience of the staff. It defines the technical skill and experience skill of the team members and clients.
7. Physical and workforce environment. It describes the daily works and physical environments of the projects.

Select the host project. The host project which is chosen represents the type of projects that usually performs by the organization. It is to ensure that the results from the evaluation are applicable to more than just the trial project. Host project is defined through organization profiles. Since the organization profile that the research chosen is typically an organization that uses an online solution through the web, the host project will use a web project as a host project. This organization profile is connected with state variables to define the detail characteristics of the project. Table 3 provides the state variables for the controlled and experiment projects.

Table 3. State variables Example

State variables	Controlled Project	Experiment Project
Application area	Line of Business App.	Line of Business App.
Framework that used	GSD process + UP method	GXP framework
Business Type	Manufacturing	Software
Scale of projects	10000-18000 LOC	10000-18000 LOC
Complexity level of problems	Low	Low
Quality and experience of the staff	4 years experiences average	2 years experiences average
Physical and workforce environment	High pressure with overtime	Casual without overtime

Application area is defined as a kind of software that being developed, which are business application. The kind of software is adopted and extended from Productivity rates for common project types (McConnell, 2006). Scale of projects and complexity levels of problems also derived from pre-projects estimation based on use case metrics and user stories that aligned with productivity rates for common project types.

Minimize the effect of confounding. The casual problem in the feature analysis case study is a subjectivity of the result. The degree of subjectivity is been based on judging method against some “evaluation criteria” which are identified subjectively based on context dependent in case studies. There is also a problem of inconsistency in scoring between evaluators, the different evaluator will give different score interpretations in a different way. The various score has to be collated and compared to decide the relative order of distinction frameworks. Furthermore, certain features may attract higher average scores than others because an assessor possibly will appreciate them better and be more able to be familiar with the framework. Another problem of feature analysis is that hundreds of features may be acknowledged, and it will become time consuming to evaluate.

Those confidence problems are constraints the evaluation to decide the level of confidences such as.

1. The case study is held by one organization with several assessors, although it will context dependent and subjective. It will give the same baseline of the confidence result.
2. The case study will use the software development life cycle (SDLC) process as a base path to achieve the result. Using SDLC will give real work experience regarding the framework.
3. The case study will limit the response variable that already stated in above.

In order to give an objective result, the research makes several assumptions that described in the next section. The research makes several assumptions regarding the risk that will be handled in the evaluation process. These assumption purposes are to limit the divergences of the researches object. There are three assumptions that assumed in the research which are.

1. People assumptions. The case study is done by the equal people who have the same level of experience at least two years as the developer, technical skills in the web framework they build, and proficiency skill in terms of teamwork.
2. Product assumptions. The case study is done by the equal project complexity, including the same framework that used to build the product. It means that the projects have a same project types (web application), same project purposes (business application), and same amount of user stories.
3. Process assumptions. The case study is done by the equal base process model. The experiment uses a global software development situation as a background process that adopted in different method and tools.

Those assumptions are the boundary and limitation for the case study is done in terms of objectivity.

3.2 Evaluation Plan

Evaluation plan must identify all issues to be addressed so that the evaluation runs smoothly. This including the evaluator, the data gathering procedures, and the measures needed for the analysis.

Evaluators are the entire person or team which responsible in running exercise for an evaluation. The team is responsible for.

1. Preparing the evaluation plan.
2. Identifying the candidate framework
3. Identifying each distinct group of user population
4. Eliciting the requirements of each user
5. Identifying the features to be assessed
6. Organizing the assessment whereby each of the frameworks.
7. Collation and analyze scores.
8. Preparation of the evaluation report.

The evaluation involves the clear separation between evaluation staff and experimental subjects. Evaluation staff should work as independent assessor, which evaluates the experiment outside the subject's environment. Therefore, evaluation staff should not technically join the project in order to concentrate with the evaluation process. On the other side, experimental subjects should not know that their works are evaluated as a case study. This approach is to make sure the subject to work and do the evaluation naturally with fewer human risk side effects like novelty or expectation effect as stated in section 2.2.

Data's gathering procedures. Data's gathering procedures are started by composing the evaluation form. Evaluation form should represent explicitly or implicitly what states in the evaluation required properties. Evaluation form should also describe the experiment plan and baseline rating. After the evaluation form is done, the evaluator should works as follows.

1. Identifying the subject by looking at the potential user who holds specific roles in the project.
2. Identifying the tasks to be performed by the experimental subjects using the framework.
3. Organizing any exercise or support for experimental subjects.
4. Running the required measurements according to the case study plan
5. Evaluating the forms and preparing the evaluation reports.

Case study minimizes the effect of individual assessor differences since it will use the real-life experience of project development. However, the problem is whether the result of the experiment will scale up in the different project. This problem will be covered by using a baseline rating in the semantics of measurement section.

The measures needed by the analysis. The measurement is done based on the case study execution. The case study will gain both quantitative and qualitative information. The quantitative information provides direct calculation and comparison between proposed and existing framework in terms of numbers. The qualitative information provides additional information how the subject of the experiment uses the properties that discussed in the evaluation. The measurement is done through these steps.

1. Evaluator joins in both projects as a quality control or assessor team member.
2. Evaluator examines the execution of the projects by following SDLC and project execution.
3. Evaluator submits the reports by filling the assessment's sheet for each project.

3.3 Evaluation Execution

This section provides detail information the execution of the evaluation based on the measurement steps which are described in section 3.2. Those steps are joining the team, following the SDLC execution, and submitting the reports. The SDLC steps that proposed in the research work as figure 2.

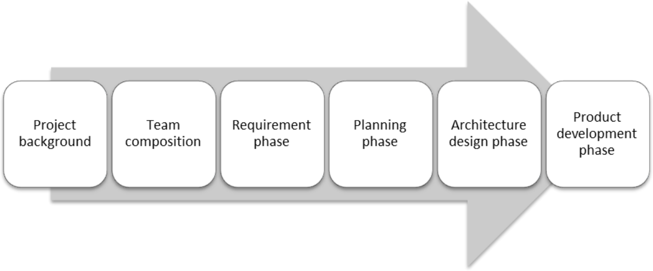


Figure 2. Evaluation execution block diagram

Project background discusses the attributes of the evaluation case studies. The research chooses project type, environment, project length, team members, framework adoption, web framework engine, platform that used for the project, and amount of software features. Those attributes should be at least same in several things in order to create a feasible comparison between case studies.

Team composition discusses the team model that implemented in the case studies. Team structure leads the mechanism of collaboration and communication between sites. In this step, the research should identify the team structure, team roles, and team communication workflow.

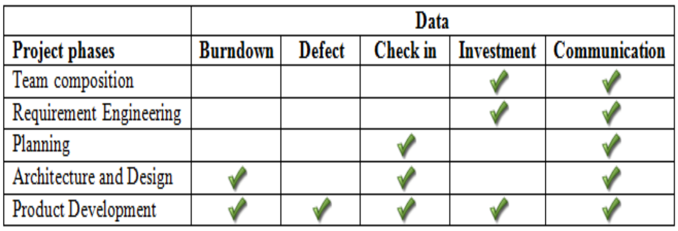
Requirement phase discusses to gather the market intent, the vision of the product, and the business purposes that are covered by the product. This phase identifies team action to fulfill this phase. It includes the artifact and the process that done by the team.

The target of planning phase is creating bundles of the features that will be developed by the team. These bundles can then be prioritized through an agreement between the client and the team. The research identifies what the team does in the planning phase.

This phase focuses in creating a design of the software, including the user interface design and software architecture design. The research identifies what the team does in the design and architecture phase.

Product development phase is the roughest phase in the software development cycle. This running application and the source code are the main output of this phase. Table 4 describes the portion of response variable that gathered in evaluation SDLC phase.

Table 4. Data acquired in case studies phases



	Data				
Project phases	Burndown	Defect	Check in	Investment	Communication
Team composition				✓	✓
Requirement Engineering				✓	✓
Planning			✓		✓
Architecture and Design	✓		✓		✓
Product Development	✓	✓	✓	✓	✓

4. Evaluation result sample

There are five response variables as a result in the researches, which are communication pattern, investment cost, check-in operation, defect rate, and burndown rate.

Communication patterns discuss the kind of patterns that happen in the overall project communication. The communication patterns discuss multimodal communication that happens in the both projects such as.

1. Email communication. It measures the amount of the email that happens between the client and development team. For the evaluation purposes, only unique email that counted in the evaluation. For an example if the manager sends same email to all the team and there are seven separate copies that the email is still counted as one email only.
2. Phone. It measures how many hours that spends for personal phone call (between peer) or conference phone calls.
3. Video conferences. It measures how many effective hours that spend for video conferences certainly it is excluding setup and configuration testing.
4. Instant messaging. It measures how many effective hours that spend for instant messaging, including private chat or conference chat sessions.
5. On site meeting. It measures how many hours that spends for onsite meeting, including the travel hours and others.

Table 5 provides the samples of communication measurement for case studies. The organization can calculate the detail of communication and see a balance between direct communication and indirect communication. The better communication provides the better result in project collaboration and cooperation.

Table 5. Communication measurements for case studies

Communication Type	Unit	Controlled	Experiment
Email	item	127	227
Phone	hour	180	40
Video Conference	hour	13	0
Instant Messaging	hour	2	113
On site meeting	hour	192	27

Investment cost discusses the framework investment that needed by the project. Investment cost includes several fix costs and variable cost such as.

1. Integrated development environment tools (IDE tool). The IDE tool is a combination of compiler, debugger, and application designer. Visual Studio, Net beans, and Eclipse are the sample of the IDE.
2. Developer component. Developer component is the third party component that purchased for developer productivity.
3. The CASE tool. The CASE tool is the software engineering software that helps the team to develop diagram, chart, and others engineering representation. Visual paradigm, enterprise architects, or Visio are the sample of the CASE tool.
4. The project management tool. The project management tool is software to manage and track project progress. Microsoft project is the sample of project management software.
5. Travel cost. Travel cost calculates the total of expands that used for the travelling budget.
6. Communication cost. Communication cost estimates the communication expands like the internet, phone, or short messages investment.
7. Hardware cost. Hardware cost estimates the cost of development hardware cost. Development hardware is a set of hardware that needed for development only. Development server, notebook to build the codes is the samples of the hardware cost.

These cost is obtained through an informal audit. The evaluator estimated the cost by seeing several proof of receipt, existing item, and investigated the prices of the development assets.

As mentioned, Burndown patterns are a set of pattern that displays the progress of the project in terms of remaining works versus the iteration of time. The interesting part is that the analysis of the burndown chart can expose various indicators on how the team is undertaking the plan and what can they do to improve further. The burndown pattern answer this following question

1. How good the team planning?
2. How well is this team executing against the planned stories in iteration?
3. Is this team self-organized and are they working in unison as a "team"?

Both two projects are side by side compared within their burndown chart. The data burndown chart is gathered from the manual calculation based on the project planning and requirements documents. The research counts the requested features, the works constraint, and others technical works that needed to be done. Figure 3 shows the burndown sample between two projects. The slight slope provides better project productivity.

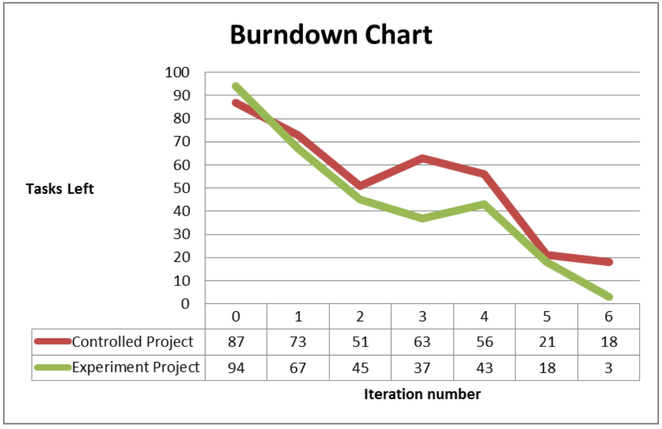


Figure 3. Burndown Chart

Defect rate discusses how many defects that exist in the both projects. Defect can be categorized as follows.

1. Bug or program error that happens when the software is running.
2. The misleading feature or wrong features that built by the team.

3. The integration or algorithm error. The errors that make the software work, but it gives a different result than expected.

Based on these categories, the evaluation counts the error from the project. The source of the information comes from feedback log, error notification through email, and compiler error counter. Figure 4 shows the sample of defect rate. The less defect rate shows the better project management to handle the defect.

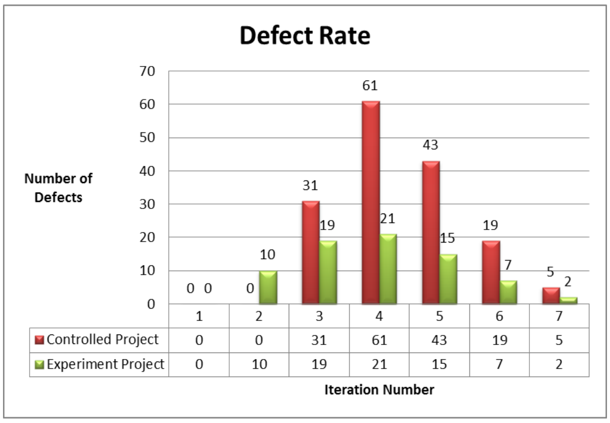


Figure 4. Defect rate chart

Check-in operation discusses how many changes that happen in the released codes and how the team makes a better improvement of the codes. The revision of the system, the fresh new build, and the new version of the software is clear identification of the check-in operation.

Check-in operation is measured through several approaches. The evaluator uses a milestone counter, nightly build, and revision logs that note by the team which are.

1. The successful build of the system and uploaded into the development system.
2. The minor revision of the project such as adds news features; fix several bugs and new facelift of the user interface.
3. The milestone of the project, since every milestone of the project shows several improvement features of the project.

Figure 5 shows the check-in operation for the both projects. The higher check in rates of operation shows the continued improvement of the project.

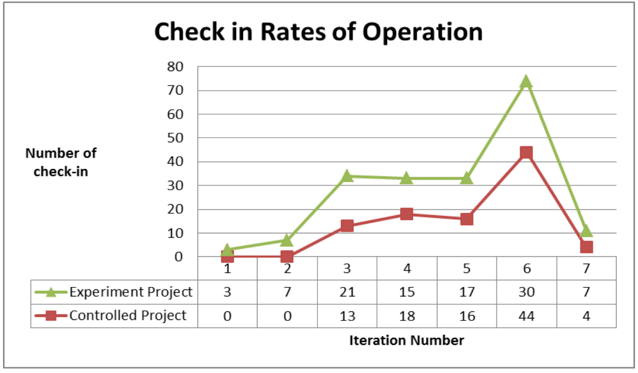


Figure 5. Check in rates of operation

The evaluation results provide a summary for the organization to choose between two or more frameworks. In a simple manner, organization can compare the result between them and choose which one that provides better productivity and efficiency.

The challenge that usually happens to be evaluating the framework without sacrifice the organization productivity. Some of the organization has no interest to create the experiment project, and others have no interest to adopt the new framework to their real project because its risks. In this kind of situation, the research advises the team to evaluate the framework based on the real and low risk project. Evaluating the real and low risk project can capture the picture of the team when using the framework.

5. Conclusion and Future Work

In software engineering, evaluating a software methodology become the important thing for the organization. Organization that chooses the correct methodology will gain the benefit as long as its productivity.

This paper limits an evaluation approach to evaluate a software engineering framework. This paper chooses DESMET as a baseline of the research framework and adds several steps that proposed in this research such as follows.

1. Adopting DESMET as an evaluation preparation step. In this step state variables and case study context is chosen.
2. Following SDLC in the evaluation execution step.
3. Identifying the results based on the response variables that already defined in the preparation step. Response variables are a set of artifacts that implicitly described the performance of the project.

As a further work, the framework evaluation should be detailed with the risk that identified when adopting this evaluation and improving the efficiency the execution evaluation without leaving a risk for the case study.

References

[1] Ferdiana, R. Nugroho, E.L, Santoso, I.P, and Ashari, A. 2010. Global eXtreme Programming, a Software Engineering Framework for Distributed Agile Software Development. IJCSET 1, 3 (October 2010), ISSN 2044-6004.

[2] Kitchenham, B. A. 1996a. Evaluating software engineering methods and tool—part 1: The evaluation context and evaluation methods. SIGSOFT Softw. Eng. Notes 21, 1 (Jan. 1996), 11-14.

[3] Kitchenham, B. A. 1996b. Evaluating software engineering methods and tool—part 2: Selecting an appropriate evaluation method—technical criteria. SIGSOFT Softw. Eng. Notes 21, 2 (Mar. 1996), 11-15.

[4] Kitchenham, B. A. 1996c. Evaluating software engineering methods and tool—part 3: selecting an appropriate evaluation method—practical issues. SIGSOFT Softw. Eng. Notes 21, 4 (Jul. 1996), 9-12.

[5] McConnell, S. 2006 Software Estimation: Demystifying the Black Art. Microsoft Press.

[6] Sadler, C. and Kitchenham, B. 1996. Evaluating software engineering methods and tool—part 4: the influence of human factors. SIGSOFT Softw. Eng. Notes 21, 5 (Sep. 1996), 11-13.

[7] Sangwan, R., Bass, M., Mullick, N., Paulish, D. J., and Kazmeier, J. 2007. Global Software Development Handbook (Auerbach Series on Applied Software Engineering Series). Auerbach Publications.

Author Biographies

Ridi Ferdiana. Mr. Ridi Ferdiana was born in 1983. He is a doctoral student at Gadjah Mada University, Yogyakarta since 2008. He earned his master degree from the same university in 2006. In his professional area, he holds several professional certifications such as MCP, MCTS, MCPD, MCITP, MVP and MCT. In his daily research activities he really enjoys to learn about software engineering, business platform collaboration, and programming optimization.

Lukito Edi Nugroho. Born in 1966, Dr. Lukito Edi Nugroho is an Associate Professor in the Department of Electrical Engineering and Information Technology, Gadjah Mada University. He obtained his M.Sc and PhD degrees from James Cook University in 1995 and Monash University in 2002, respectively. His areas of interest include software engineering, distributed and mobile computing, and application of ICT in education.

Paulus Insap Santosa. Insap was born in Klaten, 8 January 1961. He obtained his undergraduate degree from Universitas Gadjah Mada in 1984, master degree from University of Colorado at Boulder in 1991, and doctorate degree from National University of Singapore in 2006. His research interest including Human Computer Interaction and Technology in Education.

Ahmad Ashari Place and date of birth: Surabaya, May 2nd 1963. Get Bachelor’s degree 1988 in Electronics and Instrumentation, Physics department Gadjah Mada University, Yogyakarta. Master degree 1992 in Computer Science, University of Indonesia, Jakarta Doctor Degrees 2001 in Informatics, Vienna University of Technology. Major Field of study is distributed system, Internet, Web Services, and Semantic Web.