Association Rule Mining for Re-planning Decisions of Software Product Releases

S. M. Didar-Al-Alam
Department of Computer Science
University of Calgary
Calgary, AB, Canada
smdalam@ucalgary.ca

ABSTRACT

Well-planned projects might end unsatisfactorily if changes in software release plan are not controlled and managed. While process control detects potentially risky changes, re-planning adjusts a current plan to changing reality. Association rule mining can be used to discover interesting relations like associations, correlations etc. among different attributes of large databases. We investigate and evaluate how association rule mining can be used to influence re-planning decisions of software product releases utilizing extracted information (i.e. rules and process factors) from legacy data.

Proposed approach *Dyna-H2W**, encompasses association rule mining based methods to find multiple process factors and associated rules responsible for project success or failure. Inspired by statistical process control, these rules are continuously monitored throughout the project. Any rule violation triggers a warning followed by an expert-based investigation for root cause analysis behind the warning. Investigation might suggest initiating re-planning as a reactive action.

This paper exclusively focuses on empirical analysis of the proposed approach. Two controlled experiments along with three case study designs are presented in this paper. Preliminary experiments and comparative analysis against existing re-planning techniques suggest *Dyna-H2W** is capable of generating product of higher *value* and *quality*. Additionally proposed approach increases stakeholders' *level of trust* on the re-planning method.

Categories and Subject Descriptors

Management, Experimentation

General Terms

Management, Measurement, Performance,

Keywords

Empirical analysis, association rule mining, software release planning, re-planning strategies, legacy data, decision support.

1. INTRODUCTION

Release planning is a complex, wicked problem [19] of assigning features to subsequent releases considering technological and business objectives and constraints. Release planning is an ongoing activity that doesn't stop when the first plan has been created. Changes during the implementation of a release must be controlled and managed. Re-planning [21] and associated process control [12] are crucial to achieve successful releases accommodating these changes. Process control monitors whether defined rules are observed and whether essential process factors (e.g. effort consumption, feature change requests etc.) are within defined value ranges [12]. If rules are violated or factors are out-of-range, corrective action has to be initiated. The importance of process control has been signaled in both industry and academia [12, 25]. Re-planning is the process of modifying a plan to incorporate changes. Release re-planning techniques support deciding when re-planning should happen, how to perform re-planning, and which features should potentially be eliminated, added or replaced while re-planning. Nevertheless, comprehensive research is still required as re-planning software release is not yet mature enough [21].

From multiple case studies Danesh et al. [22] deduced that experience organizations prefer to enrich their existing products instead of planning for a complete new one. Extracted rules and process factors from legacy projects can be used to support re-planning decisions and process control for upcoming projects. This potential source of decision support is still unutilized in existing re-planning methods. Objective of this paper is to investigate how mining legacy data (e.g. association rule mining technique) can be used to extract rules and process factors to influence re-planning decisions of software product releases. Association rule mining [1] is a popular and well-researched data mining method, capable of extracting significant information (i.e. rules and process factors) from available legacy data.

Existing re-planning method *Dyna-H2W* [4] is capable of dynamic re-planning (i.e. dynamically deciding when to re-plan) in consideration of multiple process factors. In this paper we propose an extension of this approach labeled as *Dyna-H2W** to provide decision support in re-planning utilizing association rule mining. Association rule mining based methods extract rules and process factors from existing legacy data. Discovered rules are further filtered, combined and prioritized [4] into a monitoring rule set. This rule set and associated process factors are utilized in continuous monitoring of the software development process. Process monitoring is inspired by Statistical Process Control (SPC) [12]. Any rule violation (e.g. deviation of the process from expected performance) triggers a warning. Human experts analyze root causes of the warning and find the best way to manage root causes. Detail of this approach is presented in Section 5.

In this paper, we exclusively focus on configuring, calibrating and evaluating *Dyna-H2W** with empirical analysis. We followed GQM (Goal-Question-Metric) [6] approach. Our goal is to investigate and evaluate how association rule mining can support re-planning decisions utilizing rules and process factors extracted from legacy data and achieve better release plan compared to existing re-planning methods. Three case studies along with two controlled experiments address four key research questions presented in Section 4. Study designs for all five empirical studies are presented in Section 7.

Empirical studies are followed by comparative analysis against existing re-planning techniques e.g. *H2W* [21], lightweight re-planning [5], *Dyna-H2W* etc. Release plans are evaluated in terms of release *value* and *quality*. Preliminary results from these analyses suggest that *Dyna-H2W** is capable of creating products of higher *value* and *quality*. Additionally it increases stakeholders' *level of trust* on the re-planning method. Our contributions in this paper are- 1) Utilizing association rule mining to support re-planning decisions, 2) Dynamic re-planning based on multiple process factors. 3) Software process control in consideration of multiple process factors and associated rules. 4) Evaluation of proposed approach in real world empirical studies.

The paper is organized as follows. Section 2 presents background methodology. Related Work is presented in Section 3. Section 4 lists our research questions. Solution approach *Dyna-H2W** is presented and illustrated with aid of hypothetical example in Section 5. Some important metrics are defined in Section 6. Five empirical study designs are presented in Section 7. Finally, Section 8 presents the conclusion and future work.

2. BACKGROUND METHODOLOGY

2.1 Strategic Release Planning

Strategic release planning is an optimized feature selection process. *F* is a set of *n* features (i.e. *F = f(1),f(2), …, f(n)*). Vector *x* = (*x(1)… x(n)*) represents a strategic release plan where the i-th component specifies the release to which feature *f(i)* is assigned (illustrated in Eq. 1 & 2). In this paper five alternative strategic release plans are created using EVOLVE II method [21], and its implementation in ReleasePlanner™ [11]. Human experts are responsible to select one plan from five diversified best plans.

$$x(i) = k \; if \; feature \; f(i) is \; offered \; in \; release \; k \qquad (1)$$

$$x(i) = 0 \; otherwise \qquad (2)$$

2.2 Operational Release Planning

Operational release plan is generated on top of a strategic release plan. Features are implemented through a sequence of tasks performed by available resources. Operational release planning defines the order of implementing tasks to achieve minimized release time or maximized release value. Available resources are assigned to different tasks based on their skillset and availability. In this paper we perform operational planning with RASORP method [20] and its implementation in ReleasePlanner™ [11].

2.3 Re-planning

Re-planning [5] is the complex problem of re-assigning features in next release to accommodate changing reality along with achieving higher release value. It encompasses both strategic and operational release planning. Accommodating changes can be performed broadly in three ways- i) Re-allocating resources among tasks ii) Re-adjusting planning goals and iii) Re-planning features (i.e. eliminate, add or replace features). Re-planning is an effective way which considers incoming feature requests along with accommodating other changes.

2.4 Association Rule Mining

Association rule mining is a well-researched and well-accepted data mining method capable of discovering interesting relations like associations, correlations etc. among different attributes of large databases [1]. Association rules are generated based on co-occurrences of attributes in databases. If $\alpha \Rightarrow \beta$ is an association rule, α, β are respectively antecedent and consequent of this association rule. *Support* and *Confidence* respectively measures statistical significance and certainty of an association rule. Details regarding association rule mining are available in Section 5 & 6.

3. RELATED WORK

Accommodating change requests is crucial for a software project to survive with changing reality. But actions required to accommodate changes are not integrated in release planning methods [21]. Typically change requests are ignored until completion of one release and re-considered in the next release [26]. A key approach to detect potentially risky changes is process control. Outside the software engineering community Statistical Process Control (SPC) has become a part of the development process. While authors in [9] reported software characteristics refraining direct use of SPC in software projects, E. F. Weller reported practical applications of SPC [25] in software project management. In proposed approach *Dyna-H2W**, we utilize a customized process control technique inspired by SPC.

Re-planning is an effective way to accommodate changes in release plan. A lightweight re-planning method was initially proposed in [5]. It performs comparison among incoming features and old features using AHP (Analytical Hierarchy Process) and chooses the best feature set using greedy approach. *DynaReP* accommodates changes through re-allocating resources [3]. *H2W-Pred* [2], performs static re-planning in consideration of two process factors (i.e. effort and quality of release plan). *Dyna-H2W* is capable of dynamic re-planning and process monitoring in consideration of multiple factors. This approach is presented and evaluated in [4].

Association rule mining reveals software defect associations and predict their corrective effort requirement in [23]. In [18] association rule mining detects potential causes for software failures. Association rule mining can support re-planning decisions as well through mining legacy data. But this potential source of decision support is still unutilized in existing re-planning techniques [2, 3, 5]. In this paper we extend *Dyna-H2W* approach to support re-planning decisions with rules and process factors extracted from legacy data. Association rule mining based methods are utilized to extract information (i.e. rules and process factors) from legacy data.

4. RESEARCH QUESTIONS

Association rule mining based methods extract information (i.e. rules and process factors) from legacy data. We extend current planning approach *Dyna-H2W* to support re-planning decisions utilizing this extracted information from legacy data. To accomplish our research goal we evaluate extended approach *Dyna-H2W** using empirical studies. Four key research questions (RQ) addressed in this paper are-

- **RQ1:** Does "when to re-plan" decision influence how and what to re-plan decisions along with re-planning results?
- **RQ2:** Does applying multiple process factors in re-planning generate release plan of higher value and quality compared to applying single process factor?
- **RQ3:** Does applying extracted information (i.e. rules and process factors) from legacy data in re-planning increase value and quality of release plan compared to RQ2?
- **RQ4:** Does applying extracted information (i.e. rules and process factors) from legacy data in re-planning increase stakeholders' trust on re-planning method compared to RQ2?

Figure 1 demonstrates how research questions (RQ) and empirical studies (ES) fit in the re-planning problem. Re-planning is a part of software development process that encompasses three key questions When? How? & What? to re-plan. RQ1 & ES1 explore the "When to re-plan" problem to understand its influences on How and What to re-plan decisions along with re-planning results.

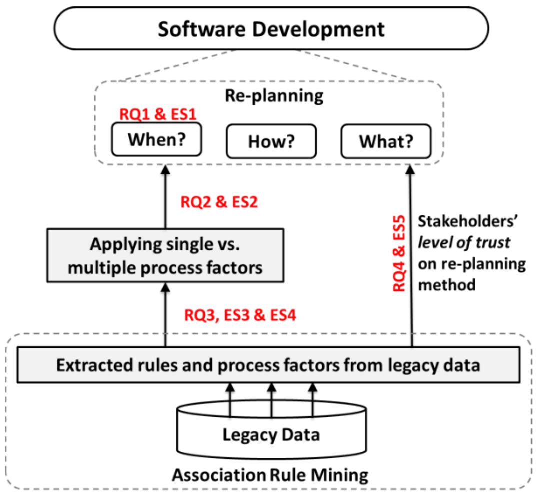

Figure 1. Relationship among RQs and Empirical studies

Applying multiple process factors instead of single might improve release decisions. RQ2 & ES2 investigate into it. RQ3, ES3 & ES4 investigate how applying extracted information (i.e. rules and process factors) from legacy data in re-planning further improve release decision of RQ2. RQ4 & ES5 investigate whether applying extracted rules and process factors in re-planning improve stakeholders' trust on re-planning method

5. SOLUTION APPROACH

This section proposes an extension for *Dyna-H2W* approach to provide decision support in re-planning and associated process control utilizing association rule mining. To provide this decision support we extract rules and process factors from legacy data utilizing association rule mining based methods. Proposed approach *Dyna-H2W** is presented in Figure 2 and explained in following sub-sections. The top frame of Figure 2 presents analysis of legacy data. The middle and bottom frame respectively present the monitoring process and continuous release plan update through re-planning. Detailed process steps of *Dyna-H2W** approach is presented in Table 2.

5.1 Analyzing Legacy Data

Legacy data refers to massive, complex collection of historical project data that generally requires data mining support to be understood and utilized. In this paper we consider project specific historical data as legacy data for analysis. Mining legacy data can extract helpful information to predict future trends and behaviors in software project management. For example mining bug repositories helps to understand how bugs are created, related with components and can help predict risky components.

5.1.1 General Method

To illustrate the general method, we consider Jazz development project. Jazz development data from the Jazz repository [15] is considered as existing legacy dataset. Details of this dataset are presented in subsection 7.4. Association rule mining based methods are applied to extract rules and process factors responsible for or related to success or failure of project or project components. Revealed association rules are combined into compound rules. A prioritization methodology ranks these rules and includes them in the monitoring rule set of software project management.

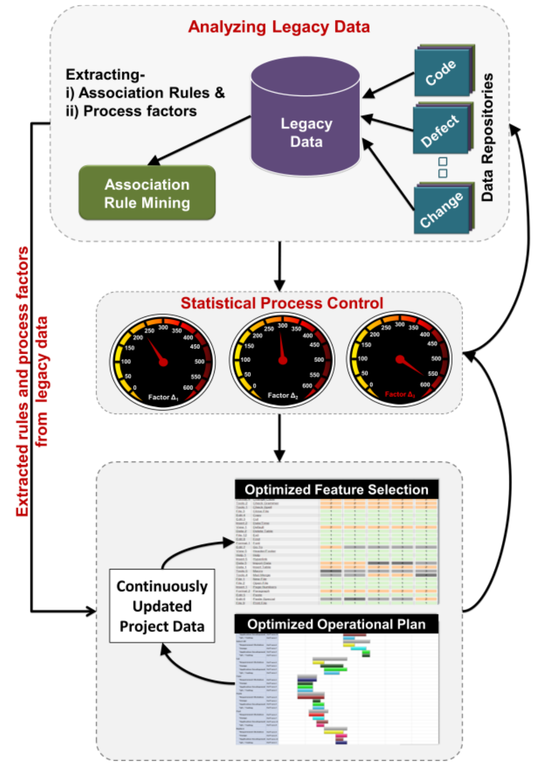

Figure 2. Proposed solution approach Dyna-H2W*

5.1.2 Association Rule Mining

Association rule mining [1] is a popular data mining method that discovers rules from large databases. It is a data driven process and the rules are generated based on co-occurrences of different attributes in database. Rules must satisfy minimum *support* and *confidence* value to be considered as valid. *Support* measures the usefulness of an association rule while *Confidence* measures the strength of an association rule. These measures are later utilized in prioritizing these rules as well. Both measures are defined in Section 6.

Table 1. Association rule examples

Association Rules	Sup port	Confi dence
(1) 70% development completed and 20% product quality degraded from plan => final product quality target failed.	50	60
(2) 5% workforce increased => solved 10% deviation in effort consumption.	10	15
(3) incoming features are 7% of initial feature set => no impact on re-planning	50	40
(4) 15% decrease in testing workforce => product quality degraded 30%.	40	50
(5) incoming features are 15% of initial feature set => product quality degraded 20%	50	40

If $\alpha \Rightarrow \beta$ is an association rule, α, β are respectively antecedent and consequent of this association rule. I is a set of n attributes called *Items*. α, β are called *Itemset* where both α, β are subset of I (i.e. $\alpha, \beta \subseteq I$). But within α, β nothing is common (i.e. $\alpha \cap \beta = \phi$). We perform association rule mining in three steps 1) Frequent *Itemsets* that satisfy minimum support value are discovered. 2) Association rules are established based on co-occurrences of discovered *Itemsets* in database. 3) Valid *r*ules (i.e. satisfy minimum support and confidence value) are filtered from all generated rules. Table 1 lists hypothetical example of some rules to demonstrate their possible appearance in reality. A data mining technique (in our case association rule mining) alone might not be capable enough to establish such concrete rules as presented in the hypothetical example. Experts are responsible for establishing rules utilizing retrieved information from mining.

5.1.3 Combining Rules

Rules are updated in two stages before they are utilized in monitoring process.

- **Stage1:** Combine discovered and valid association rules into compound rules towards success or failure of project or project components.
- **Stage2:** Prioritize rules and include them in monitoring rule set based on their *support* and *confidence* values.

For example association rule mining on legacy data extracted five rules presented in Table 1. Rules with minimum support and confidence value (e.g. 20, and 30 respectively) are considered valid. 4 rules (i.e. Rule 1,3,4,5) out of 5 in Table 1 are marked as valid. Rule 1 is directly associated with failure of the project. Others are indirectly related with success or failure of project components.

Rule 1, 4 and 5 are grouped into one compound rule *Rule-C*. Due to lack of relations with other rules Rule 3 is not combined into any compound rule. Combined *support* and *confidence* values for *Rule-C* are calculated utilizing Eq. 3, 4. α is considered as antecedent and consequence of these rules. While combining *n* rules *support* and *confidence* value of the compound rule is respectively the summation of *support* and *confidence* values of *n* rules which gets combined.

$$combined\ support = \sum_{i=1}^{n} Support(\alpha_i \Rightarrow \alpha_{i+1}) \qquad (3)$$

$$combined\ confidence = \sum_{i=1}^{n} Conf(\alpha_i \Rightarrow \alpha_{i+1}) \qquad (4)$$

5.1.4 Prioritizing Rules

Filtered and updated rules are prioritized based on their *support* and *confidence* values respectively. For example, rules are initially prioritized based on their *support* value. If two rules of same *support* value are encountered *confidence* value is applied to prioritize them. While high *support* value reflects high usefulness of the rule, high *confidence* value implies high certainty of the rule. Prioritized rules are added into the monitoring rule set.

5.2 Project Monitoring and Re-planning

This section of proposed approach follows its predecessor *Dyna-H2W*. Prioritized monitoring rule set and associated process factors create re-planning criteria prior to the project development. For strategic plan x_1 and corresponding operational plan $y_1(x_1)$ creation of a re-planning criteria considering process factor *effort* (i.e. effort consumption) is as follows.

For strategic plan x_1, *m* different resource types are available denoted as r_i $(i=1...m)$. At time *t* for resource type r_i most recent estimated effort consumption is $effort_{est}(t,r_i)$ and actual effort consumption is $effort_{act}(t,r_i)$. According to extracted information from legacy data maximum allowed difference (i.e. absolute difference) between estimated and actual effort consumption for resource r_i is λ_i. Eq. 5 presents the corresponding re-planning

Table 2. Detailed steps of *Dyna-H2W**

Step-1: Start ($t=0$, $i=1$)
Step-2: Pre-process legacy data & find association rules (i.e. Association rule mining)
Step-3: Filter, combine & prioritize rules to a valid rule set *R* with *n* rules $R_1, R_2,..R_n$
Step-4: Determine associated process factors $\Delta_1, \Delta_2,.., \Delta_m$ & re-planning criteria $RC_1, RC_2, ..., RC_k$ in respect of rule set *R*.
Step-5: Optimized feature selection (i.e. strategic planning)
Step-6: Selection of the most appropriate plan x_i (out of a set of optimized and diversified plans generated in Step 5)
Step-7: From the selected plan x_i, generate optimized operational plan $y_i(x_i)$ for the upcoming release.
Step-8: During implementation at any time t, monitor in respect of process factors $\Delta_1(t)$, $\Delta_2(t)$ and $\Delta_m(t)$
Step-9: As long as $t < T$ and one of the warning criteria $RC_1, RC_2, ...,RC_k$ is violated, a warning is triggered. Go To Step 10. Otherwise Go To Step 12.
Step-10: Re-plan and update project data in terms of process factors $\Delta_1, \Delta_2,.., \Delta_m$
Step-11: $i = i + 1$ and Go To Step 5
Step-12: Stop

criteria (RC_{effort}). RC_{effort} is continuously monitored throughout the project and a warning is triggered at time $t=t^*$ if RC_{effort} (i.e. Eq.5) is violated.

$$\Delta(t^*) = abs\big(effort_{est}(t, r_i) - effort_{act}(t, r_i)\big) < \lambda_i \qquad (5)$$

Comprehensive example in consideration of multiple process factors (e.g. quality, effort consumption and features change request) along with real world case study is available in [4]. Violation of re-planning criteria triggers warning for *out-of-control* situation. *S*ituations that fall outside the maximum allowed value ranges of a monitoring rule referred as *out-of-control* situation. Such situation requires experts for root cause analysis behind the warning. Re-planning can be performed in three different intensity categories e.g. (a) Manual re-plan, (b) Lightweight re-plan [5] (c) Optimization-based re-plan (e.g. using EVOLVE II). Expert can manage *out-of-control* situation using one of these re-planning categories. After completion of one re-plan monitoring process starts over again.

6. DEFINITION OF METRICS

This section defines some important metrics used in this paper to collect data from empirical studies and perform comparative analysis against existing re-planning techniques.

Value: Value of a release plan implies the level of stakeholder's satisfaction on the generated plan. Value is measured in SFP (Stakeholder Feature Points). In presence of *m* planning criteria (*p*) and corresponding weight (*pw*), *q* stakeholders (*s*) and corresponding weight (*sw*) and stakeholder votes (*sv*), SFP for a feature *f(n)* in plan *x* (i.e. *sfp (n,x))* is calculated in Eq. 6 below.

$$sfp\ (n, x) = \sum_{i=1}^{m} (\ p_i w \ \times \sum_{k=1}^{q} (s_k w \times s_k v)) \qquad (6)$$

Quality: Quality of a software is difficult to measure. For simplicity we evaluate quality through reliability of software. Reliability is mapped into testing resource requirements to detect and fix defects. For feature *f(n)*, Cost of Quality *CoQ(n)* is measured using Eq. 7 utilizing Cost of Conformance *CoC (n)* and Cost of Lack of Conformance *CLoC (n)* measures [8]. These measures are cost of detecting defects (i.e. integration and testing effort) and cost of fixing detected defects respectively.

$$CoQ\ (n) = CoC(n) + CLoC\ (n) \quad (7)$$

Support: *Support* measures the statistical significance of an association rule [1, 23]. Multiple *Transactions* (τ) constitute *Database* (θ). $\alpha \Rightarrow \beta$ is an association rule. *Support* of this association rule is x (i.e. $Support\ (\alpha \Rightarrow \beta) = x$) means x% of *transactions* (τ) in *database* θ contains both $\alpha\ and\ \beta$.

Confidence: *Confidence* represents the strength [1, 23] of an association rule. *Confidence* value for association rule $\alpha \Rightarrow \beta$ is $Conf\ (\alpha \Rightarrow \beta)$ calculated according to Eq. 8. In database θ *transactions* (τ) that contains α in $Conf\ (\alpha \Rightarrow \beta)$% cases also contains β.

$$Conf\ (\alpha \Rightarrow \beta) = Support\ (\alpha \cup \beta) /\ Support\ (\alpha) \quad (8)$$

Trust: *Trust* [10] value measures the degree of trust stakeholder has on the method applied to generate a release plan or re-plan. Controlled experiment in ES5 utilizes two variables to measure trust. While *Level of trust* measures the degree of stakeholders' trust in a five point scale (i.e. substantially low, low, medium, high, substantially high), *Evaluation of Trust Level* measures the variation in degree of trust in a five point scale (i.e. substantial increase, increase, equal, decrease and substantial decrease).

7. EMPIRICAL ANALYSIS

This section presents empirical study (ES) plans to answer research questions and evaluate *Dyna-H2W** approach. Three case studies (ES 1,2,3) and two controlled experiments (ES 4,5) are presented with detail design. ES1 is partially accomplished but the final analysis is still outstanding. ES2 is already accomplished and results are reported in [4]. ES 3, 4 and 5 are considered as future study plans.

7.1 Answering RQs by Empirical Studies

Empirical study plans (i.e. ES1 to ES5) presented in this section address four RQs of this paper. ES1 investigates "when to re-plan" problem addressing RQ1. ES1 performs re-planning at different timestamps (e.g. at the beginning of each iteration) within one release duration. Significantly different results achieved from each re-plan infers the impact of "when to re-plan" decision on how and what to re-plan decisions.

Applying process monitoring utilizing multiple process factors can find better time to initiate re-planning and improve re-planning decisions. *Dyna-H2W* [4] offers multiple process factor based process monitoring for dynamic re-planning. ES2 empirically analyze this approach in a real world case study along with addressing RQ2. Result exhibits *Dyna-H2W* is capable of creating release plan of higher value and quality compared to applying single factor. In this paper we propose an extension of *Dyna-H2W* approach labeled as *Dyna-H2W**. Proposed extension extract rules and process factors from legacy data utilizing association rule mining based methods to support re-planning decisions. ES3 empirically analyze *Dyna-H2W** with a case study and addresses RQ3.

We further investigate *Dyna-H2W** with ES4 & ES5. ES4 primarily addresses RQ3 and performs a controlled experiment to explore how utilizing extracted information from legacy data improves re-planning decision compared to RQ2. Controlled experiment in ES5 addresses RQ4 and explores evaluation in stakeholder's *level of trust* on re-planning method due to applying extracted information from legacy data. All five empirical study plans are presented in following sections.

7.2 ES1: Case Study 1

7.2.1 Goal

Goal of ES1 is to investigate how time to trigger *out-of-control situation* warning and initiate re-planning impact re-planning decisions. This study addresses RQ1 and reveals relationship between when to re-plan and re-planning decisions.

7.2.2 Data Source, Tools and Techniques

An open source text editor software development project is considered in this case study. This project considers 50 initial features. Four stakeholders voted features on two planning criteria (i.e. Urgency, Value) for upcoming one release. Incoming features arrive within one release duration. Resource requirements are initially available from expert evaluation. Five developers are responsible for implementing tasks corresponding to each feature.

Initial strategic and operational release plans are generated using EVOLVE II and RASORP planning methods. Release duration encompasses six iterations of equal length. Static re-planning is performed at the beginning of each iteration except the first iteration. Five re-planning timestamps are referred as very early, early, intermediate, late, and very late. This case study performs re-planning with Greedy optimization approach.

7.2.3 Incoming Features

Incoming features arrive after the project implementation has already begun. Experimental design applies randomization in number of features arrived, arrival time, resource requirements, and stakeholder votes. Different experiment trials consider number of incoming features as one of three categories- i) Heavy (1 to 20 incoming features), ii) Moderate (1 to 15 incoming features), iii) Low (1 to 5 incoming features). Feature arrivals are uniformly distributed within one release duration.

7.2.4 Study Plan and Analysis Strategy

Strategic release plan and corresponding operational plan are generated using planning methods. Each trial of this study performs single re-plan at either one of five re-planning timestamps. Numerous trials are performed to accomplish a reasonable result. Actual resource requirements varies from initially estimated requirements as project continues. Barry Boehm empirically evaluated [7] resource estimation accuracy in different phases of software development. For example, in Requirement elicitation phase estimation accuracy varies within 2x to 0.5x while in Testing phase variation narrows down within 1.10x to 0.91x. Actual resource requirements are calculated by varying initial requirements with triangular distribution within the resource estimation accuracy ranges [7] as presented in Table 3.

Table 3. Resource requirement estimation accuracy

Development phases	Variation ranges	Mode
Requirement Elicitation	[0.5,2.0]	1.5
Design	[0.67,1.5]	1.25
Development	[0.8,1.25]	1.10
Testing	[0.91,1.10]	1.00

Table 3 presents resource estimation accuracy ranges correspond to different development phases along with mode value applied in triangular distribution of this study. Results from all trials are further analyzed using Whisker-Box plot analysis in terms of release *value* and *quality*. Final analysis results are still outstanding. Expected outcome is to achieve a relationship among when to re-plan decision and re-planning results.

7.2.5 Threats to Validity

Major limitations of this study are - 1) open source software (OSS) (e.g. text editor) development does not represent all existing strategies for software development, 2) some feature dependencies had to be discarded due to consideration of one upcoming release, 3) ES1 considers static re-planning at five different timestamps. Considering more re-planning timestamps might provide even better analysis.

7.3 ES2: Case Study 2

7.3.1 Goal

ES2 empirically evaluates *Dyna-H2W* approach and addresses RQ2. Goal of this case study is to investigate how applying multiple factors in re-planning and process monitoring influence re-planning decisions.

7.3.2 Study Plan, Analysis Strategy & Results

This case study considers an open source text editor software development project. The project encompasses 50 candidate features, four stakeholders and two planning criteria (i.e. Urgency and Value). Strategic planning and operational planning are performed using EVOLVE II and RASORP planning methods respectively. This case study is already accomplished and detail process steps along with empirical analysis are reported in [4]. Detail project data are available online [14].

Comparative analysis is performed against two re-planning strategies- 1) Dynamic re-planning with single factor and 2) Static re-planning with single factor. We consider no re-planning as the baseline approach. *Value* and *quality* of release plans from all three strategies are compared and results are reported in [4]. Result demonstrates *Dyna-H2W* in presence of multiple process factors generates release plan of higher value and quality compared to other re-planning approaches.

7.3.3 Threats to Validity

Accomplished result cannot claim external validity due to underlying assumptions and simplifications. Major threats include- 1) for defect detection and fixing we only consider quality assurance (QA) effort responsible. But additional effort might be required in reality 2) though effort, value, strategic and operational plan of this study are collected from real world, QA data are synthetically generated. 3) In reality re-planning criteria might require re-adjustments within a release period. But this study considers re-planning criteria constant for one release duration.

7.4 ES3: Case Study 3

In software development single metric is not capable of predicting software failures [17]. Thus we plan to explore multiple metrics combining different repositories e.g. change histories, version histories, bug repositories, code repositories etc. Primarily we explore Jazz development data available in the Jazz repository.

7.4.1 Goal

Case study in ES3 is designed to address RQ3. Goal of this study is to empirically analyze how proposed approach *Dyna-H2W** performs in re-planning compared to existing re-planning methods e.g. *H2W*, lightweight re-planning, *Dyna-H2W* etc. This study investigates the influence of utilizing extracted rules and process factors from legacy data in re-planning and associated process control decisions.

7.4.2 Data Source, Tools and Techniques

IBM developed Jazz [15] as a team collaboration software that captures software development process, artifacts, real time changes, team communication etc. Jazz integrates software development process with team collaboration. Thus artifacts available in the Jazz repository are linked together. The Jazz repository is a well-accepted and comprehensive data repository that can fulfill our extensive demand of data. This repository has a complex database structure that encompasses more than 200 relations. Thus data extraction from the Jazz repository becomes challenging due to high speed and huge storage requirements. Moreover detection of misleading information existing in database is very difficult. From Jazz repository we reveal software metrics using the IBM Rational Software Analyzer Tool and association rules using predictive analytics tool Weka [13]. We extract five pairs of training and test data sets from available data to use them in a five-fold cross validation approach [23]. Initial strategic and operational plan are conducted using EVOLVE II and RASORP methods.

7.4.3 Study Plan and Analysis Strategy

Available Jazz development data from the Jazz repository are extracted and pre-processed to use in Weka. Association rule mining based methods reveal association rules related to success and failure of different project components of Jazz. Discovered rules are further filtered, combined and prioritized into a monitoring rule set to monitor the software development process. Any monitoring rule violation triggers a warning and human experts manage the root cause of the warning. Accomplished release plans are measured in terms of *value, quality* and compared against existing re-planning techniques e.g. *H2W*, lightweight re-planning, *Dyna-H2W etc*. Expected outcome is to generate release plan of higher value and quality by utilizing extracted rules and process factors from legacy data.

7.4.4 Threats to Validity

Major limitations exist in ES3 are- 1) The Jazz repository provides detailed data on software development. Other repositories rarely provide such detailed data. Results might vary with other repositories due to unavailability of data. 2) Association rule mining extract rules from legacy data. Further study should investigate how other data mining techniques perform in proposed approach. 3) Re-allocating resources among different tasks might impact re-planning results. To mitigate this threat we consider all resources (i.e. staffs) are equally skilled. But in reality it is hard to accomplish.

7.5 ES4: Controlled Experiment 1

7.5.1 Context

ES4 primarily addresses RQ3. Goal of this experiment is to explore how *Dyna-H2W** supported with extracted information (i.e. rules and process factors) from legacy data performs re-planning compared to existing re-planning methods e.g. *H2W* approach. Experimental design follows guideline from [16]. *Subject* i.e. 20 (twenty) undergraduate students from University of Calgary, Calgary, Canada perform this experiment in a tutorial session of *Software Process and Project Management* course. *Object* is a project extracted from Jazz development data in Jazz repository. Roughly this project will constitute of 50 features, 3 planning criteria, 5 resources and a pool of available developers.

7.5.2 Hypothesis Formulation

Following null hypotheses (i.e. H1, H2) are formulated to accomplish the goal of ES4. Expected outcome is to reject these hypotheses and possibly reveal contrary hypotheses.

- **H1:** Applying extracted information (i.e. rules and process factors) from legacy data in re-planning cannot increase value (i.e. SFP of release plan) of re-planning decisions.

- **H2:** Applying extracted information (i.e. rules and process factors) from legacy data in re-planning cannot increase quality (i.e. number of defects detected and debugged) of re-planning decisions.

7.5.3 Selected Variables

Information (i.e. rules and process factors) are extracted from legacy data utilizing association rule mining methods to support re-planning decision. *Treatment* is considered as an independent variable. Two values possible for *treatment* are - *XS* (utilizing extracted information with *Dyna-H2W** approach) and *XWS* (unavailability of extracted information with *H2W* approach). *Control Group* and *Treatment Group* respectively perform *XWS* and *XS treatment. Value* and *Quality* are considered as dependent variables. *Value* implies how valuable the plan is to the stakeholders. It is measured in SFP (Stakeholder Feature Points). *Quality* measures number of defects detected and debugged.

7.5.4 Experimental Design & Data Analysis

This experiment is performed in two parallel stages. *Object* is a real world project. According to randomized design approach [27] all 20 *Subjects* (i.e. student participants) are randomly distributed among both *treatments.* Each *treatment* is performed by 10 participants.

- **Stage1:** *Control group* perform *treatment XWS*. Extracted information from legacy data is not available for re-planning decision support.

- **Stage2:** *Treatment group* perform *treatment XS*. Re-planning and process control are performed based on rules and process factors extracted from legacy data.

Required training and tools (e.g. ReleasePlanner™ tool, experiment materials etc.) are provided to *Subjects.* Each *Subject* performs only one treatment. Release plans from both groups (i.e. *control* and *treatment*) are collected and evaluated in terms of *value* and *quality*. Considering the size of our sample (i.e. twenty) we perform non-parametric test (e.g. Mann-Whitney test [27]) to accept or reject our hypotheses. Expected outcome is to reject both null hypotheses and find contrary relations.

7.5.5 Threats to Validity

ES4 tries to mitigate threats through its experimental design and choice of real world project. Some major threats exist are- 1) *Subject* (i.e. undergraduate students) cannot equally represent project managers due to lack of experience. But additional training on the experiment can partially mitigate this threat. 2) Sample size is relatively small (i.e. twenty). This study must be performed in more complex industry environment to accept or reject current results.

7.6 ES5: Controlled Experiment 2

7.6.1 Context

Objective of ES5 is to investigate whether extracted information (i.e. rules and process factors) from legacy data increases stakeholders' level of trust on the method applied in re-planning. ES5 answers RQ4. Twenty undergraduate students from University of Calgary, Calgary, Canada perform this experiment in a tutorial session of *Software Process and Project Management* course. They are considered as *Subject* of this experiment. *Object* of this experiment is extracted from Jazz development data in Jazz repository. Guidelines from [16] are followed in designing this experiment.

7.6.2 Hypothesis Formulation

Null hypothesis *H3* is formulated to accomplish the objective of this controlled experiment. Expected outcome is to reject the null hypothesis. We will try to reveal a contrary hypothesis as well.

- **H3:** Applying extracted information (i.e. rules and process factors) from legacy data in re-planning does not increase stakeholder's *level of trust* on the re-planning method.

7.6.3 Selected Variables

Rules and process factors are extracted from legacy data utilizing association rule mining based methods (discussed in subsection 5.1). *Treatment* is considered as an independent variable. Two values possible for *Treatment* are *YS* (performs *Dyna-H2W** approach and utilize extracted information form legacy data) and *YWS* (performs H2W approach. extracted information form legacy data are unavailable). Two dependent variables are *Level of trust* and *Evaluation of Trust Level.* These values are measured in a five point scale as specified in Section 6.

7.6.4 Experimental Design & Data Analysis

Initially a survey collects information related to *subjects. Subjects* perform both treatments (i.e. *YS* and *YWS*) on same *Object* in two stages. Experimental data from these stages are collected in two subsequent surveys. ES5 experimental design encompasses two stages and three surveys as described below.

- **Survey1:** Initially this survey collects information regarding background and experience in software project management for all *Subject* (i.e. undergraduate students).

- **Stage1:** *Treatment YWS* is performed with *H2W* method. Rules and process factors extracted from legacy data are not available.

- **Survey2:** Records experimental data from stage1. Collects stakeholders' *Level of trust* on re-planning method applied in *treatment YWS*

- **Stage2:** *Treatment YS* is performed with *Dyna-H2W** approach. Re-planning and process control is performed based on process factors and rules extracted from legacy data.

- **Survey3:** Records experimental data from stage2. Collects stakeholders' *Evaluation of trust level* while applying *Dyna-H2W** approach in *treatment YS*.

Chi-square [24] analysis is a well-known statistical hypothesis test approach. It is popularly used for frequency comparison. It can detect whether a population has a specific variance or not. We further analyze our survey results and test our hypothesis using Chi-square analysis. It detects the cause behind frequency variations and thus accepts or rejects the null hypothesis.

7.6.5 Threats to Validity

Experimental design and choice of project partially mitigate threats in ES5 as well. Still major limitations exist are- 1) *Subject* (i.e. undergraduate students) are not equally experience as project managers. Thus additional training is provided to partially mitigate this threat. 2) Due to small sample size result cannot be generalized. This study should be replicated in complex industry environment to accept or reject current results. 3) Survey questions and metrics are subjective. It imposes the threat of losing meaningful objective data.

8. CONCLUSION & FUTURE WORK

In this paper proposed approach *Dyna-H2W** utilizes association rule mining based methods to extract rules and process factors from existing legacy data. This extracted information supports re-planning decisions of software product releases. Five empirical study plans presented in this paper evaluate the proposed approach. Additionally, these studies investigate how association rule mining supports re-planning decisions in terms of *value*, and *quality* of release plan and stakeholders' *level of trust* on re-planning method. Results are compared against existing re-planning techniques (e.g. *H2W*, lightweight re-planning, *Dyna-H2W* etc.). Expected outcome is to prove *Dyna-H2W** is capable of creating products of higher *value* and *quality* compared to existing re-planning techniques.

Future works should perform continual data analysis and extract information from continuous data of software projects along with existing legacy data to support re-planning decisions. Proposed studies should be replicated in more complex industry settings to accept or reject revealed results. Replicating these studies after mitigating existing threats will also help to verify current results. Additionally different data mining techniques can be applied in place of association rule mining to compare their performance in proposed approach.

9. ACKNOWLEDGEMENTS

This doctoral research is supervised by Dr. Guenther Ruhe, University of Calgary, Calgary, AB, Canada and co-supervised by Dr. Dietmar Pfahl, University of Tartu, Tartu, Estonia. This research is supported by an Alberta Innovates Technology Futures fellowship of the author.

10. REFERENCES

[1] Agrawal, R., Imieliński, T. and Swami, A. 1993. Mining association rules between sets of items in large databases. *ACM SIGMOD Record*. 22, 2 (1993), 207–216.

[2] Al-Emran, A., Jadallah, A., Paikari, E., Pfahl, D. and Ruhe, G. 2010. Application of re-estimation in re-planning of software product releases. *New Modeling Concepts for Today's Software Processes*. (2010), 260–272.

[3] Al-Emran, A., Pfahl, D. and Ruhe, G. 2007. DynaReP: a discrete event simulation model for re-planning of software releases. *Software Process Dynamics and Agility*. (2007), 246–258.

[4] Alam, S., Ruhe, G. and Pfahl, D. 2013. Decision Support for Re-planning of Software Product Releases. *International Conference on Software Engineering and Knowledge Engineering* (2013).

[5] AlBourae, T., Ruhe, G. and Moussavi, M. 2006. Lightweight replanning of software product releases. *International Workshop on Software Product Management, IWSPM'06.* (2006), 27–34.

[6] Basili, V.R., Caldiera, G. and Rombach, H.D. 1994. The goal question metric approach. *Encyclopedia of software engineering*. 2, (1994), 528–532.

[7] Boehm, B.W. 1984. Software engineering economics. *IEEE Transactions on Software Engineering*. 1, (1984), 4–21.

[8] Campanella, J. 1999. *Principles of quality costs: Principles, implementation and use*. ASQ Quality press Milwaukee, WI.

[9] Card, D. 1994. Statistical process control for software? *Software, IEEE*. 11, 3 (1994), 95–97.

[10] Du, G., McElroy, J. and Ruhe, G. 2006. A family of empirical studies to compare informal and optimization-based planning of software releases. *Proceedings of International symposium on Empirical software engineering* (2006), 212–221.

[11] Expert Decisions: *www.expertdecisions.com*.

[12] Florac, W.A. and Carleton, A.D. 1999. *Measuring the software process: statistical process control for software process improvement*. Addison-Wesley Professional.

[13] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P. and Witten, I.H. 2009. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*. 11, 1 (2009), 10–18.

[14] https://sites.google.com/site/didar522/data-repository: .

[15] IBM rational jazz: *http://jazz.net/*.

[16] Jedlitschka, A. and Pfahl, D. 2005. Reporting guidelines for controlled experiments in software engineering. *International Symposium on Empirical Software Engineering* (2005).

[17] Mockus, A. and Weiss, D.M. 2000. Predicting risk of software changes. *Bell Labs Technical Journal*. 5, 2 (2000), 169–180.

[18] Parsa, S., Vahidi-Asl, M. and Naree, S.A. 2008. Finding causes of software failure using ridge regression and association rule generation methods. *Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing,* (2008), 873–878.

[19] Rittel, H.W. and Webber, M.M. 1984. Planning problems are wicked problems. *Developments in design methodology*. (1984), 135–144.

[20] Ruhe, G. 2009. Optimized resource allocation for software release planning. *IEEE Transactions on Software Engineering*. 35, 1 (2009), 109–123.

[21] Ruhe, G. 2010. *Product Release Planning: Methods, Tools and Applications*. CRC Press.

[22] Seyed Danesh, A., Ahmad, R., Saybani, M.R. and Tahir, A. 2012. Companies Approaches in Software Release Planning--Based on Multiple Case Studies. *Journal of Software*. 7, 2 (2012), 471–478.

[23] Song, Q., Shepperd, M., Cartwright, M. and Mair, C. 2006. Software defect association mining and defect correction effort prediction. *IEEE Transactions on Software Engineering*. 32, 2 (2006), 69–82.

[24] Wasserman, L. 2004. *All of statistics: a concise course in statistical inference*. Springer.

[25] Weller, E.F. 2000. Practical applications of statistical process control [in software development projects]. *Software, IEEE*. 17, 3 (2000), 48–55.

[26] Wiegers, K.E. 2009. *Software requirements*. O'Reilly.

[27] Wohlin, C., Runeson, P., Hst, M., Ohlsson, Magnus C and Regnell, B. and Wessln, A. 2012. *Experimentation in software engineering.* Springer Publishing Company, Incorporated.