WHatver ON MAP

by

RAJASRI NANDURI

A Capstone Project Proposal
submitted in partial fulfillment of the
requirements of the degree of

Master of Science in Computer Science & Software Engineering

University of Washington
August 2, 2019

Project Committee:

David Socha, Committee Chair
Jeffrey Kim, Committee Member
Robert Dimpsey, Committee Member

1 Introduction

Software architecture refers to the fundamental structure of a software system and the discipline of creating such structures and systems [1]. A software architecture comprises of software elements, relations among them, and their properties. It serves as the blueprint for the system and not only governs software development and evolution but also determines the overall characteristics of the resulting software system [2]. Since the architecture of a software system constrains the quality attributes, the decisions taken during architectural design have a large impact on the resulting system. Choosing an architecture for a system is not straightforward. To implement a system, one has a combinatorially growing number of design options at their disposal. While there are a plethora of options, deciding on a suitable architecture design of a system lies with the desired quality attributes[3] and constraints of the use-case, because at the end, the stakeholders decide if a system is useful or usable. "Like Physics, Medicine, Manufacturing, and many other disciplines, software something somethin requires the same high level approach for evolving the knowledge of the discipline; the cycle of model building, experimentation, and learning. We must experiment to see how and when a system works. We must learn from the application and improve our understanding what works for the system" [4]. Contrary to the often simplification that all software is the same, the process, the goals and the context are all variables in the development of a software. To understand how architecture decisions are influenced by the variables, the application in this project will be implemented through an empirical process. This process of evolving the application architecture will help me study how user-feedback motivated goals and process deviate the architecture from what is predicted without feedback.

2 Goals

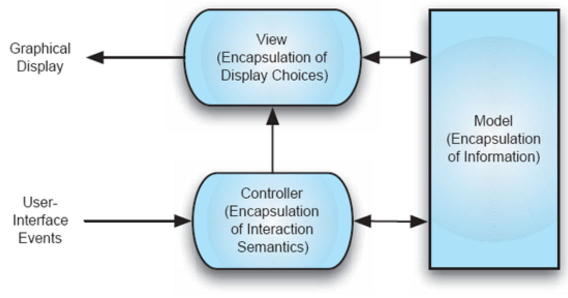
As presented in the Introduction section, there is a need for documenting the process of creating an emerging architecture and understanding how the system architecture is impacted with feedback. Through this project, I will implement an interactive React Native Mobile application application, News on map. This application will initially have a basic set of features like displaying news for the desired locality in an interactive manner and ability to share the news outside the application. The minimum viable product implemented will be presented to users for feedback in regular weekly iterations. The architectural decisions taken on the basis of the feedback is compared against the architectural decisions predicted without the knowledge of user-feedback. The feedback is collected and documented in weekly iterations. Through this empirical process, the application will be evolved into a detailed, and more functional system with high user-acceptance. The main goal is to build an emerging news on map application and studying how often does the expected architectural decisions of what would be good for the application turned out to be accurate. The news on map application will be useful for those interested in getting news specific to a locality and the process will help me understand the empirical process of implementing an emerging system with the help of regular feedback iterations.

3 Method

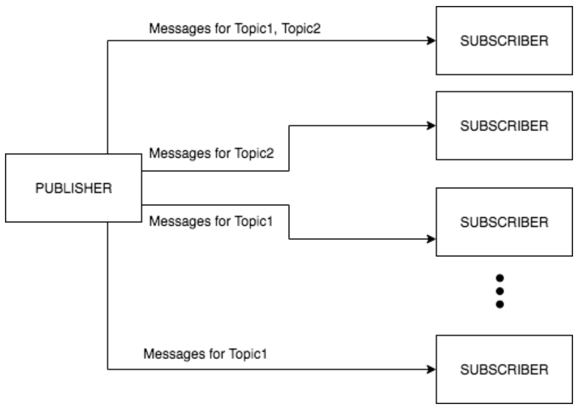
Every application needs architecture. Different sets of stakeholders may deem different sets of design decisions principal, resulting in varying needs of software quality characteristics. Even with high levels of disclosure, due to the differences in individual abilities and creativity, development of a software solution can vary from person to person. One of the objectives of Empirical Software Engineering is to gather and utilize evidence to develop applications with high user acceptance. This process provides the means to gather and disseminate evidence in order to support the claims of efficiency or efficacy of a architecture decision. Experiences and lessons learned from empirically assessing software architecture represent a valuable means of improving the application. For this project's news on the map application, I will be conducting an empirical process for formulating architecture decisions and understand how these decisions change based on feedback.

3.1 Minimum Viable Product:

The first implementation of the application will be an MVP with a basic feature set – the application user will be able to see the news content for a desired locality, the application user will be able to query news for any location. With the guidelines provided in [5] [6] to construct architectural designs in a traditional manner, I have decided on a preliminary architecture for this approach. The application will have front-end and a back-end and it will be a react native mobile application, which will be implemented using the Model-View-Controller architecture pattern, see Figure 3. This pattern is widely known for its application in programs with a Graphical User Interface. Using this pattern for controlling the data in an interactive maps channel provides a structure for the application at a higher level. In order to receive news content for the desired locality, this approach will use an implicit invocation architectural style, Publish-Subscribe. Through this style, subscribers register to receive specific content. Publishers maintain a subscription list and broadcast messages to subscribers either synchronously or asynchronously, see Figure 3. Communications between the server and the client happen using the REST architectural style.


Graphical Display
View (Encapsulation of Display Choices)
Model (Encapsulation of Information)
User-Interface Events
Controller (Encapsulation of Interaction Semantics)

(a) Model View Controller



(b) Publish - Subscribe Architecture Style for implicit fetch of news for topics subscribed

Figure 1: Preliminary architecture decisions for MVP

3.2 Empirical Process:

User an empirical process control, I will construct and implement an incremental architecture. I will collect the user requirements and feedback in weekly sprints at the end of which I will present the implemented product incremental to the users. The users will then provide feedback on the increments delivered and this process is repeated. Based on the feedback and insights gathered, I will make architecture decisions to implement the user desired increments. These increments will be granular and deduced to the smallest possible deliverables for the week. I will maintain a journal to keep track of the architecture decisions made, why they were made, what factors influenced that decision, and what other alternatives were considered before making the decision. Before the start of each iteration, I will also write down a predicted architecture of what decisions derive maximum value for the current implementation. This documentation of decision process and reasoning will help for future references and can be used to understand if and how the empirical process architecture deviates from the predicted architecture. Through this process, I will be able to study how I as an architect think in comparison to how users think about and perceive the application. The Figure 2 shows the process flow for this approach and the figure **??** show how architecture decisions and user feedback will be documented.

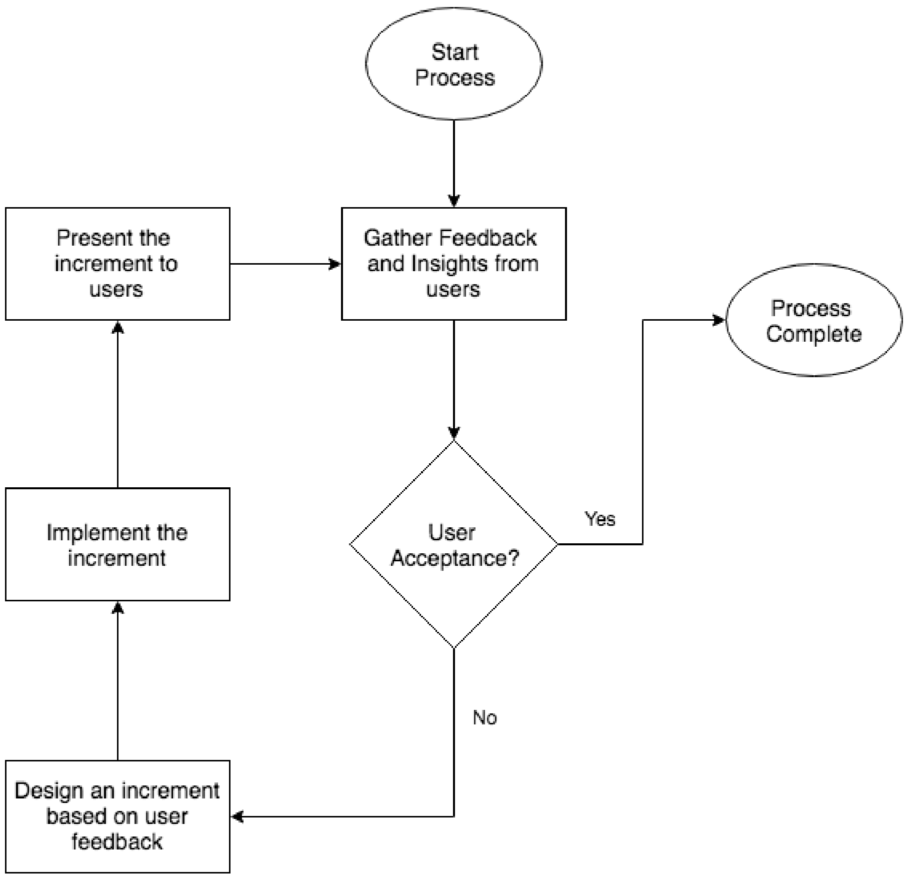
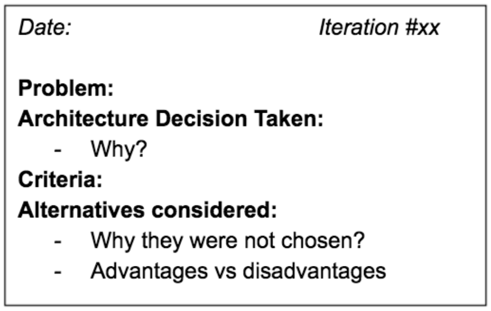

Start Process
Present the increment to users
Gather Feedback and Insights from users
Process Complete
Implement the increment
User Acceptance?
Yes
No
Design an increment based on user feedback

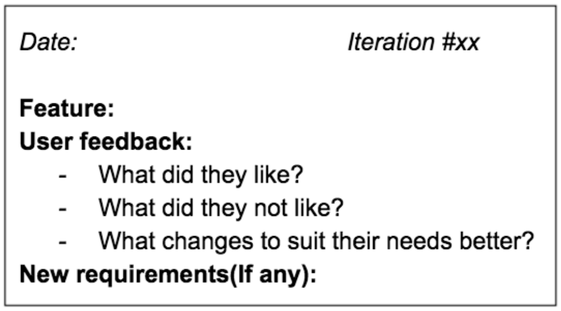
Figure 2: Empirical Process Flow


Date: *Iteration #xx*

Problem:
Architecture Decision Taken:
- Why?

Criteria:
Alternatives considered:
- Why they were not chosen?
- Advantages vs disadvantages

(a) Architecture Decisions


Date: *Iteration #xx*

Feature:
User feedback:
- What did they like?
- What did they not like?
- What changes to suit their needs better?

New requirements(If any):

(b) User Feedback

Figure 3: Documentation of architecture decisions and user feedback

In order to track the iterations and product backlog history, I will be using Trello. I will write unit tests for the increments delivered through a sprint cycle. I will be using Github for version control and the application with increments developed in each cycle will be deployed through Circle CI.

Risks: One of the probable aspects that can hinder the success of this project is insufficient user feedback. Lack of critical feedback for the architecture decisions can also be a hindrance.

4 Criteria

The following criteria can be used to evaluate project success:

Architecture Implementation:

- The application supports users to view the new content specific to a locality.
- The application supports sharing the news content in channels outside the application.
- The content on the application is presented on an interactive map.
- The application has the feature set derived from the user feedback.

Empirical Process:

- User feedback and insights collected through out the process are documented.
- Architecture decisions and interesting questions in making those decisions are documented.
- A structured iteration process is followed throughout the project.

5 Tentative Plan

The key deliverable for the project can be split into three parts. First part would be the requirement gathering and implementation of a minimum viable product. Second would be evolving the application through regular user-feedback iteractions and finally, studying the process followed for developing the application. Below is the tentative plan for the project:

Quarter	**Milestones**	**Duration**
Fall 2019	Requirement Gathering	1 week
Fall 2019	Formulate architecture a minimum viable product	1 week
Fall 2019	Implementation of the minimum viable product	5 weeks
Fall 2019 Winter 2020	Incremental design and implementation through user feedback. In Weekly cycles	9 weeks
Winter 2020	Analysing the empirical process followed	2 weeks

Table 1: Tentative Plan

References

[1] P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, P. Merson, R. Nord, and J. Stafford, *Documenting Software Architectures: Views and Beyond*, ser. SEI Series in Software Engineering. Addison-Wesley, 2010.

[2] N. Medvidovic and R. N. Taylor, “Software architecture: Foundations, theory, and practice,” in *Proceedings of the 32Nd ACM/IEEE International Conference on Software Engineering - Volume 2*, ser. ICSE ’10. ACM, 2010. [Online]. Available: http://doi.acm.org/10.1145/1810295.1810435

[3] B. J. ”Jansen, A., “Software architecture as a set of architectural design decisions,” *5th Working IEEE/IFIP Conference on Software Architecture (WICSA05)*, 2005.

[4] V. R. Basili, *The Role of Experimentation in Software Engineering: Past, Present, and Future*. University of Maryland, 1996.

[5] R. N. T. Eric M. Dashofy, Nenad Medvidovic, *Software Architecture: Foundations, Theory, and Practice*. John Wiley Sons, 2009.

[6] “Methods for software architecture,” https://www.sei.cmu.edu/research-capabilities/all-work/display.cfm?customel_datapageid_4050=21328, accessed:2019-07-25.