**Software Development Process Model**

Developers Inc.

**Assignment # 1**

**Comp3520**

Jagjit Singh Bilkhu - T00533766

Fahad Murtaza - T00527466

# Table Of Contents:

[**Executive Summary:**](#_fvsmg86z1hds) **3**

[**Business Idea:**](#_c4uvubp9ujay) **4**

[**Development Process:**](#_xiebahs5qyq) **5**

[Requirements Gathering, User stories and infrastructure setup:](#_ajdr9ibo7g3c) 6

[Manual test case and task creation:](#_1ujpi5pbn1ui) 6

[Implementation/Coding Cycles:](#_jv2nfsnz2i4e) 7

[Maintenance:](#_fbu7lq3dx4ws) 7

[**Conclusion:**](#_37y2jjoymzqk) **8**

[**References:**](#_5lhn8sw0svnm) **9**

# Executive Summary:

Developers Inc. is a mobile app company with a vision to help small businesses in various fields to adapt their business models into using a mobile median as a backbone for their operations. Our company focuses on developing and delivering mobile solutions to provide cost-effective and robust solutions to the needs of our customers from different business fields. The applications allow businesses to widen the reach of their marketing and customer engagement strategies and also digitize their operations to improve efficiency and effectiveness and also reduce overhead costs.

To develop and implement our app solutions, we will focus on a combination of scrum agile development with bits of extreme agile programming. This would allow us to break down the projects into smaller tasks and accomplish them based on their priorities in sprint cycles as sorted by the customers. It would allow us to build a great relationship with the customers, as they would see on- time delivery of increments and we would get feedback on the project progress.

# Business Idea:

Mobile apps are nowadays a necessary requirement for any business to engage with their consumers, and grow the target customer market with attractive adverts and promotions. Businesses can also use this technology to digitize their internal operations to improve efficiency and effectiveness of employees and reduce overhead costs. Such solutions require to be designed, implemented and released in a short period so that they can catch the attention of the rapidly changing markets and so that they can be effective for the purpose they were created.

Example of some Applications solutions:

1. Food ordering and delivery
2. Warehouse inventory management
3. Employee time, task tracking and payments
4. Restaurant serving and menu display
5. News Reporting and updates

For the purpose of this report, we will focus on applying our development process towards developing a mobile app that displays a restaurants menu and also allows servers to take and record table orders digitally and send them to the kitchen to be prepared. Also, it allows servers to take payments from customers that have been served via NFC or credit card and can email receipts to customers or print them if needed.

We use a generic CMS that would be used to store sample and actual data that the customers would like the mobile apps to display. This CMS will help reduce the time spent when setting up the project structure and would allow part of our team to focus on gathering requirements and user stories from the restaurants and servers that.

We would use JIRA issue tracking system to manage sprints, sprint backlogs, user stories, tasks, bugs and defects during the sprint cycles.

# Development Process:

Our development process is a custom mixture of Extreme Programming and Scrum Agile development. From extreme programming, we would adopt to using User Stories, refactoring code and pair programming.

User stories help to define the user requirements in the context of the user and allow us to have a clear understanding of the operation and expected outcome with the customers. User stories will be created through interviews, questionnaires, and observation of restaurant staff: servers, managers and kitchen staff to make them relatable to how the app will be used.

Refactoring would be used in the latter stages of development where the focus would be on fixing nontrivial code bugs, improving the performance, cleanup and readability of the code and components.

Scrum development we would adopt the iterative approach to develop the system by dividing tasks into small groups that are sorted by priority as needed by the customer.

Our process focuses having close customer involvement in the early stages of development to get requirements, design system structure, create test cases and estimate the sprint cycles and their velocity.

Once these stages are completed, the level of customer involvement required would drop and would mainly be needed for acceptance testing and validation. During the development cycles, the coding team does intense implementation and manual testing to cover as many requirements as possible. In the latter stages, the focus would be diverted to improving the system structure e.g. by automating test cases, increasing server.

## Requirements Gathering, User stories, and infrastructure setup:

The development process would begin by getting the project requirements and user stories from the customer and are entered into the product backlog from which we would setup of the development infrastructure, for example, the CMS, APIs, required libraries and frameworks.

For example, we will be getting the restaurant menu, pricing, table layouts and arrangements, server attributes, kitchen staff needs and many other sample data from the customers. These will be used to design and modify the CMS, databases, and format of data communication to be used e.g. JSON or XML

The initial user stories and sample data would ease designing and identify all positive and negative test cases and also allow us to have a clear understanding of what the customers expect the application to do. These will be entered and recorded into the JIRA system and will be tracked throughout the development process and altered if any changes are needed.

While a team is working on getting user stories from the clients, another would be refining sample data that is provided and would start blueprinting the software infrastructure. These blueprints will be used to create a paper prototype to show the customer the overall flow and operation of the application.

## Manual test case and task creation:

Once user stories and requirements are verified, they will be broken down into test cases. Using these test cases, tasks are created and are then sorted in order of priorities by scrum masters. The tasks are then distributed through the team of developers, who are given specific instructions and deadlines using the prioritized list of tasks generated from the test cases.

For the initial part of development, we will be using the manual test cases to do acceptance and negative testing to find bugs and defects in the implementation of the software before releasing. During each sprint, the bugs and defects from the current and previous sprint’s will be prioritized and fixed accordingly while new implementations and features are also being added.

## Implementation/Coding Cycles:

With test cases and tasks created, the actual process of implementing the code would be done. Tasks with a higher priority will be implemented first and tested manually against the test cases and any bugs or defects found will be recorded. As bugs are entered and recorded, they are prioritized to be fixed in the same sprint or the next. The sprints would ideally run for 4 weeks of pure implementation and testing 2 days or nightly builds to ensure that the testing team has the latest build to work on.

Customers will be involved in the development cycle sprints but won't be needed as closely involved as in the first two stages of the process. Their involvement will allow us to validate the progress of the project and to make sure that the requirements are met and done in time.

Once the major tasks and requirements have been covered, the focus would be given to automating the manual test cases to save time when testing. The automation tests would allow the testing teams to do more acceptance, integration and ad hoc tests to make sure the application is stable and has few hard to recreate defects. Relating to our project, some tasks could be to create menu screen, create payment screen, create order display screens.

## Maintenance:

After all requirements and functionalities have been achieved, the focus would be given to maintaining the backend of the application i.e. the databases, web servers, and also to fixing any bugs that are existing in the application and improving the application. Refactoring of the code would be done to end with clean, readable and organized code that can later be generalized for other application development projects as well.

# Conclusion:

We believe that the combination of Extreme Programming and Scrum Agile development would allow us to be flexible and dynamic to the customer’s requirements and structure of operation. The process allows us to be more flexible and dynamic to make changes to the system according to any change in requirements by the customers done during the iterative cycles of development when implementing the project.

Also, this will allow us to get regular feedback and validation from the customers in the cycles. Another benefit of using this process is that during maintenance, we get a chance to improve the code and make it more abstract and generic such that it can be used in the future for other similar applications that we develop thus saving time for us in the future.

Since customers may not be available at all times, having them involved during the setup process would greatly help to establish a stable foundation from which to develop the system. As development progresses, the customers would be involved as they wish s the project won't be very dependent on them. during the development stages, if changes or new features are requested, then more user stories and tasks can be created.

A weakness of this process is that it will require a very high level of customer involvement in the initial stages of the setup which may not be possible. This may result in slow setup and can delay the other stages in the process. Another weakness is that in this process, we rely heavily on setting up the infrastructure correctly in the initial stages. This can be hindered due to incorrect planning, incorrect sample data, wrong explanation by users and many other reasons. But this can be combated with the iterative development cycles where corrections and changes can be made concurrently with implementation of the user stories.

# References:

Tutorialspoint. “Software Testing - validation testing ” Software Testing Dictionary

<https://www.tutorialspoint.com/software_testing_dictionary/validation_testing.htm>

Wikipedia, “Scrum (software development)” - 20 January 2018

[https://en.wikipedia.org/wiki/Scrum\_(software\_development)](https://en.wikipedia.org/wiki/Scrum_(software_development)/)

Mount Goat Software “[Differences Between Scrum and Extreme Programming](https://www.mountaingoatsoftware.com/blog/differences-between-scrum-and-extreme-programming)” - Mike Cohn - April 6, 2007

<https://www.mountaingoatsoftware.com/blog/differences-between-scrum-and-extreme-programming>

VersionOne -”Agile Programming Best Practices” [VersionOne](https://www.versionone.com/) > [Agile 101](https://www.versionone.com/agile-101/) > Agile Programming Best Practices - date unavailable

<https://www.versionone.com/agile-101/agile-software-programming-best-practices/>