



CS 413

Software Engineering Project Management

2018-2019 Fall Semester

CAR MARKET

SOFTWARE PROJECT MANAGEMENT PLAN

Project Manager: Kübra Nur Güzel

Pınar Bayata

Kübra Nur Güzel

Arif Can Terzioğlu

Ömer Berk Uçar

1. Overview	3
1.1 Project Summary	3
1.1.1 Purpose, Scope, and Objectives	3
1.1.2 Assumptions and Constraints	4
1.1.3 Project Deliverables	5
2. Project Organization	5
2.1 Internal Structure	5
2.2 Roles and Responsibilities	6
3. Managerial Process Plan	7
3.1 Startup Plan	7
3.1.1 Project Effort, Cost, and Schedule Estimation	7
3.1.2 Staffing Plan	9
3.1.3 Training Plan	10
3.2 Work Plan	11
3.2.1 Main Activities (WBS)	11
3.2.2 Schedule Allocation (Gantt chart)	13
3.3 Control Plan	13
3.3.1. Requirements Control Plan	13
3.3.2. Schedule Control Plan	13
3.3.3. Budget Control Plan	14
3.3.4. Quality Control Plan	14
3.3.5. Reporting Plan	14
3.4 Risk Management Plan	15
3.5 Project Closure Plan	17
4. Technical Process Plan	18
4.1 Process Model	18
4.2 Methods, Tools and Techniques	18
4.3 Infrastructure Plan	19
4.4 Product Acceptance Plan	19
5. Supporting Plans	19
5.1 Configuration Management	19
5.2 Quality Assurance Plan	20
5.3 Process Improvement Plan	20
6. Additional Plans	20
Appendixes	21

1. Overview

1.1 Project Summary

Car Market is a candidate to become the ideal portal for any car-trading businesses. Car Market aims to add new features to the system where not any other rival provides in order to stand out and be appealing to users. Users create their own profiles in order to do some advertisements with the purpose of selling their own cars. Users will be able to follow some posts and get notifications in case of a price decrease. Searching through posts can be done easily with the use of tags, filtering and sorting. A car can be bought directly with the price fixed by the owner or an online auction can be held. User would have three options if he/she decided on a car; they could directly accept the price fixed by the owner, they could try to make some deals through direct messaging or the last option is by giving a bid and wait to be the highest bidder and win the auction. After the buying/selling process finishes, users will be able to rate and comment on another user they did the dealing with. This will be helpful for further trading by the same users since this will be counted as the “reliability rate” of the seller. Alongside with the “bidding” feature of the systems, users can also offer car exchange. If both users have at least one car in their account, they might consider exchanging, if there is an extra cost to one of the cars that extra will be offered as well as the trade procedure. Users can be categorized as “buyers” and “sellers” by the system. Buyers can create a “wishlist” so that Car Market can make some suggestions to the user according to their needs. Also, depending on their search history and the cars they are generally interested in, the system might also put up some recommendations to have a better user experience. Sellers can put up posts of the autos they are planning to sell, they also need to provide some contact information, or they might choose the option to only make direct-messaging available. As Car Market, we will also offer some professional drivers to deliver the purchased car to the new owner. With the notary work, the seller can give authority to the drivers and trading businesses can be done very easily.

1.1.1 Purpose, Scope, and Objectives

Purpose: With Car Market, we tried to analyze other rival products in the market and came up with some features that we thought would create a trustworthy, easy and better trading environment for regular users and we will try to make it as user friendly as possible to make any person be able to use the product without any background experience.

Scope: Car Market aims to develop a user-friendly car trading platform with added features for the system to be able to compete with the already existing ones on the market. Car Market application is for posting cars to sell, search for cars to buy and communicate between buyers and sellers in order to create a trustworthy trading environment.

Objective: Car Market is providing a new car-trading platform with enough new features and appeal so that we will get the significant share of the market. We decided to create certain goals to make this more tangible.

- Whether a users sales will increase with this system or not
- Whether the usage for the application can increase day by day
- Whether changes can be made for the system in the future or not
- Whether the feedback from the users will be positive or not

1.1.2 Assumptions and Constraints

Assumptions:

- Sellers will not struggle when using the advanced search, the system with all of its components will work perfectly.
- The system will be trustworthy when counting the honesty scores of the sellers.
- There will be improvements according to the given feedback of the system.
- A seller who is in trouble can reach Call Center members without any problem.
- Each user will have a access to the internet and supported browser on their pc/phone.
- All browser that users use will support HTML standards.

Legal Constraints:

- There will be a legal constraint about prices of cars. For example, only Turkish lira can be used in order to sell and buy a car, if the car belongs to Turkey.
- A user cannot buy a car from another country.
- There will be an auditing constraints to meet minimum government or financial standards.

Sustainability Constraints:

- The system will be adaptable for future updates on different platforms.

Security Constraints:

- Users of client software can define and change the password in order to prevent other users' access.
- Users of server software can define and change the password in order to prevent other any outside access.

- There will be some security constraints in order to prevent customers' data when they type their card number and password.

Reliability Constraints:

- For getting more accurate data, the photos of the cars will be checked.
- Tested and trusted software will be used.
- The drivers to deliver the car will be hired professionals in order to create a trustworthy purchasing environment.

Implementation Constraints:

- For a collaborative contribution, GitHub platform will be used.
- The software will have server-client architecture.

Technical Constraints:

- Response time will be less than five seconds.
- The software will support all browsers including mobile and mini browsers.
- The server will support intense online traffics.

1.1.3 Project Deliverables

- Project Charter - already delivered
- Software Requirements Specification Report - already delivered
- Low-Level Design Report
- High-Level Design Report
- Implementation
- Functional Test
- User Acceptance Tests
- Pilot Usage (Project Demo)
- Final Report
- Effects of the Project

2. Project Organization

2.1 Internal Structure

In this project, the project team will adapt the projectized organization type after comparison with a strong matrix organization. Since there are no functional managers in our company, the characteristics of our organization are closer to projectized organization. By this way, Sponsor of our E-commerce system will have a communication channel to the

project. The project manager can also keep in touch with Sponsor more efficiently. In other words, The Project Manager will be responsible to the investor company.

The team will have a faster response mechanism and decision-making processes. Also, the team will be focused only on one project at a time, the concentration will be higher. During the flow of the project, team members periodically will give reports directly to the manager. He/she will be responsible for the performance and progression of team members.

2.2 Roles and Responsibilities

Responsibility	Roles	Participant
<ul style="list-style-type: none"> Communicate team members and Sponsor about the project all the time Make necessary decisions and changes on the project Set up teams Receive reports Enable the information and share among member and customer 	Project Manager	Kübra Nur Güzel
<ul style="list-style-type: none"> Design the overall system architecture Divide the project into smaller parts with respect to reusability and sustainability Check the code to ensure the quality of the design 	Software Architect and Back-end developer	Arif Can Terzioğlu
<ul style="list-style-type: none"> Writing well prepared, testable, efficient code by using best software development practices Integrating data from various back-end services and databases Integration of UI elements developed by front-end developers with server-side 	Web Application and Back-end Developer	Ömer Berk Uçar

<ul style="list-style-type: none"> • Responsible for a user-friendly website with regarding the system • Developing wireframes and task flows based on user needs • Conducting user research and testing 	Front End Developer and UX Specialist	Pinar Bayata
<ul style="list-style-type: none"> • Collaborating with the front-end developers and other team members to establish objectives • Integration of user-facing elements developed with server-side logic • API Integration and back-end services 	Backend Developer	Staff (1 People)
<ul style="list-style-type: none"> • Create test scenarios • Apply tests and report the results to manager 	Software Tester	Staff (2 people)
<ul style="list-style-type: none"> • Tackle user acceptance testing in a timely and organized fashion 	User Acceptance Tester	Staff (2 people)
<ul style="list-style-type: none"> • Creates a database system and ensure the safety of data • Guarantee the continuity and accessibility of the database • Responsible for the performance and capacity of the database 	Database Analyst	Staff (1 People)

3. Managerial Process Plan

3.1 Startup Plan

3.1.1 Project Effort, Cost, and Schedule Estimation

Assuming as a team, we are all senior and highly skilled manager, designers and developers to finish this project, we should have a solid cost estimation and schedule according to our past experiences and variants. In the case of the best parallelization, we ought to achieve most of the work simultaneously by regarding the dependencies between phases.

The initial planning phase of the project which contains planning, time and cost management, risk management planning and etc. will take at least **6 weeks**.

System Engineering, requirement elicitation and overall architecture of the system will take the priority since other tasks will be dependent on them. Requirement elicitation will require at least **2 weeks**. Here, we assumed that low and high-level architectures (in system engineering) are dependent on customer requirements. Therefore, system engineering will take about **3 weeks** together with all dependencies.

System architecture design, requirement elicitation and software analysis, wholly, considering the scale of the project and similarities to the older projects in terms of reusability, will approximately take at least **5 weeks** in the case of start to start working mechanism.

We will start UI design of the system parallel with software requirements. Since system and database design are dependent on low and high-level architecture and both take **2.5 weeks**, the software design phase will be about **5 weeks** in total in a parallel working manner.

In further, the project will be divided into modules and parts decided by the developer team. After the separation, even though the developers will be able to work independently, they should need to be in sync by communication and daily meetings in order to implement and test the necessary functions. In accordance with the reliance and wait for the other modules, assuming a total of **2 weeks** lost in communication and wait is appropriate. One of the crucial parts which are the front end, back end and web application coding will need around **20 weeks** in order to create a robust and secure system based on calculation. Calculation for this part is done in the following way. We assumed that the average line of codes is around **4.5KLOC** from past projects and constants are $a = 2.5$, $b = 1$, $c = 1.5$ and $d=0.5$.

$$\text{Effort} = a * (\text{size})^b = 2.5 * (4.4)^1 = 11.25$$

$$\text{Duration} = c * (\text{Effort})^d = 1.5 * (11.25)^{0.5} \approx 5 \text{ mounts} = 20 \text{ weeks}$$

$$P = \text{Effort} / \text{Duration} = 11.25 / 5 = 2.25 \approx 3 \text{ men (Developers)}$$

Implementation will need incremental testing and redesign this is why it will approximately. By including unit testing and unit debugging, software implementation will take at least **6.5 mounts**. Whole testing processes (testing plan, unit testing, component testing, system testing and database testing) will take at least **2.5 months** (some parts will be in parallel with implementation). After the testing phase is finished and the product is ready to release,

deployment and release activities will begin. This phase will take at least **8 weeks**. Complete software engineering phase will take at least **8.5 months**.

According to these estimations and Gantt chart, delivery of the application will approximately take **10 months** from the start of development until the first release stage.

For the cost estimation, if we use average salaries for developers around **3000€ monthly** and again assuming the development team members will be helping out in working on other modules as their initial work is finished, during the software engineering phase, $8.5 * 3000 * 11 = 280.500€$. Project management part will cost around $1.5 * 3000 * 1 = 4500€$. System engineering and deployment activities will cost around **2400€** and **6300€** respectively.

For the other expenses, we can assume the database will be on a cloud data centre and will not be in-house, this will cost around **1200\$ per month** (AWS standard storage fees with the average of 50 TB of data, region: Frankfurt). Website domain name and web hosting fees will be around **2000€ per year**. Remaining tools such as Adobe Photoshop and JIRA will cost **230\$ and 120\$** respectively **per year**.

At least 5 employees will be working on the future updates, configuration and maintenance of the current system, so this will create a cost of **15,000 €** cost per month after initial release if we assume each will earn **3000€ monthly**.

Note: Change in requirements may affect these approximations. In this case, re- estimations will be performed.

3.1.2 Staffing Plan

In Car Market project, we will develop a web-based application therefore highly skilled web developer and back-end developers are needed. Car Market will store information about user credentials, payment methods and include a large number of money transactions thus, we need an experienced and highly skilled database analyst for a secured database system. Our project needs a strong foundation and divide-and-conquer analogy so that it is advantageous to have an architect who will design system architecture and divides the project into smaller parts in a sustainable manner. UI/UX developers will be able to design user-friendly and know well the principles of good design with the innovation. Since this product will go into the market, we must ensure that it will satisfy user criteria via testers. Lastly, the project needs a project manager who will communicate the team members and sponsor about the project at the end of a project. She will also make decisions and changes in the project. There were 9 staff who are responsible for different areas of the project in the

project charter phase. However, we have predicted that one web developer may not be enough so we will hire one back-end developer for server-side operations and we have redistributed some roles. In the team, the total number of staff is 10 currently.

Roles	Number	Phases
Project Manager	1	Project Management Training Software Analysis Deployment & Release Supporting Activities
Software Architect	1	System Eng. Requirement Elicitation Software Analysis Software Design
Back-end Developer	1	Software Implementation Deployment & Release
Web Application Developer	1	Software Implementation Deployment & Release
UI/UX Specialist & Developer	1	GUI Design / Implementation
Software Testers	2	Testing
User Acceptance Testers	2	Testing
Database Analyst	1	Database Design

3.1.3 Training Plan

Communication and motivation are really important for the project. It is a well-known fact that communication between team members have a strong impact on the development process of the project. We decided to arrange seminars about communications. Group members may be unfamiliar with e-commerce. So, we will give an introductory seminar on e-commerce systems. On the technical side, since we have an experienced and highly skilled staff, we will only arrange online courses in the case of need. These courses may include PHP or HTML- CSS. Naturally, it is necessary to encourage all group members to participate in these seminars to produce a successful project. Additionally, according to the

evaluation of the manager in case the new members join the group, seminars can be arranged for them in order to make the group complaint.

3.2 Work Plan

3.2.1 Main Activities (WBS)

The Work Breakdown Structure (WBS) is shown below:

1.0. Car Market

2.0. Project Management

- 2.1. Planning
- 2.2. Requirement Management
- 2.3. Time and Cost Management
- 2.4. Risk Management
- 2.5. Quality Management
- 2.6. Resource Management
- 2.7. Documentation

3.0. Training

- 3.1. Defining development Tools
- 3.2. Deciding familiarities of Tool
- 3.3. Team members' orientation

4.0. System Engineering

- 4.1. Planning of technical background
- 4.2. Defining system requirements
- 4.3. Design high-level system architecture
- 4.4. Design low-level system architecture

5.0. Software Engineering

- 5.1. Requirement Elicitation
 - 5.1.1. Software requirement specification
 - 5.1.2. Customer requirements
 - 5.1.3. Development team requirements
- 5.2. Software Analysis
 - 5.2.1. Assigning task to a team
- 5.3. Software Design
 - 5.3.1. GUI Design

- 5.3.1.1. User Interface Design
- 5.3.2. System Design
 - 5.3.2.1. Unit Design
 - 5.3.2.2. Component Design
 - 5.3.2.3. System Design
- 5.3.3. Database Design
 - 5.3.3.1. Creating Tables
 - 5.3.3.2. Constraints
 - 5.3.3.3. Creating relations
- 5.4. Software Implementation
 - 5.4.1. Software Unit coding
 - 5.4.1.1. Back-end Services
 - 5.4.1.1.1. API Integration
 - 5.4.1.1.2. Server Requests / Functions
 - 5.4.1.2. Login System
 - 5.4.1.3. Search System
 - 5.4.1.3.1. Filtering
 - 5.4.1.4. Recommendation System
 - 5.4.1.5. Posting Mechanism / Page
 - 5.4.1.6. Bid System
 - 5.4.1.7. Payment
 - 5.4.1.9. Driver /Delivery Tracking System
 - 5.4.1.8. Message System / Page
 - 5.4.1.10. Main Page
 - 5.4.1.11. Advertisement Page
 - 5.4.1.12. Users' Posts / Favorite Posts Page
 - 5.4.1.13. Purchased Items / Delivery Page
 - 5.4.1.14. Notifications Section
 - 5.4.2. Unit debugging
- 5.5. Testing
 - 5.5.1. Testing Plan
 - 5.5.2. Unit Testing
 - 5.5.3. Component Testing
 - 5.5.4. System Testing
 - 5.5.5. Database transaction testing
- 6.0. Deployment & Release Activities
 - 6.1. Deployment Management

- 6.2. Configuration Management
- 6.3. Change Management
- 6.4. Verification & Validation
- 6.5. Acceptance Testing
- 7.0. Supporting Activities
 - 7.1. Tracking System
 - 7.2. Quality Assurance
 - 7.3. Documentation Tools
- 8.0. Closure
 - 8.1. User Manual
 - 8.2. End-user training

3.2.2 Schedule Allocation (Gantt chart)

This section provides a scheduling relationship among work activities with time-sequencing constraints and concurrent work activities. Duration and due date of each workgroup, predecessors and successors of each endeavour and their objectives to others and resource over dissemination are provided. Gantt chart can be found in the Appendix.

3.3 Control Plan

3.3.1. Requirements Control Plan

Since the incremental process model is used in this project, requirements are not finalized until the end of the project. Therefore we will use Requirement Add Request document for adding new requirements to the system. Details can be found in 3.3.5.

For requirements control traceability, prototyping and reviews techniques will be used. We will use requirements traceability matrix(RTM) to keep track of requirements. When a requirement changed or added or its status is changed then the RTM table will be updated by the project manager.

Requirement Control Report will be documented when a requirement is failed or changed. Details can be found in 3.3.5. Project manager and team members decided how they will response to requirement failure.

3.3.2. Schedule Control Plan

The schedule of the project is determined at Project Charter report, therefore, milestones and required signs of progress and outputs at the milestones are predefined.

Actual progress is assessed according to planned milestones and expected deliverables. Schedule performance index is used as control mechanism to measure progress. Moreover, we will use the critical path method to determine the most crucial activities and regularly check them to guarantee critical path activities are not falling behind.

To control schedule progress team members will meet and share the progress of their current activities each week. Also, at milestones, team members will come together and document the Schedule Control Report. This report also used when actual progress cannot follow the planned schedule. Details can be found in 3.3.5. Schedule will be rearranged and lacking work will be aimed to completed as soon as possible. If some changes are needed in schedule plan or activities it will be reported as a Schedule Change Request. If the request is accepted then the schedule will be changed as stated in the Schedule change request.

3.3.3. Budget Control Plan

The budget of the project is determined at Project Charter report, therefore, it is a fixed resource and during the project, it is strictly controlled. Project manager will meet with company managers and sponsors once every two weeks regularly and give update about budget. The budget is measured as earned value and project manager will calculate cost performance index and present it to other stakeholders. For cost performance index, planned value, actual cost, earned value, cost variance will also be calculated. To calculate the earned value the progress of the project, quality of deliverables, evaluation of milestones will be used as criteria.

When actual cost is different than planned cost and at the end of milestones, Budget Control Report will be documented. Based on the positive or negative deviation in planned budget adjustments will be adapted if necessary.

3.3.4. Quality Control Plan

All deliverables produced during the project are evaluated according to the quality control plan. Quality of the product will be assessed with feedbacks from the stakeholders. According to quality assessment results, the developer team decided on necessary changes and update the product. When the quality of the product is not acceptable according to quality assurance, Quality Control Report will be documented.

3.3.5. Reporting Plan

Three reports are prepared before the project development. These three reports are Project Charter, Software Requirement Specification, and Software Project Management

Plan respectively. Reports are prepared according to IEEE standards and project schedule. Other than these reports, the following status and control reports will be used during the project:

The Requirement Add Request is used when a stakeholder wants to add a requirement into the system. The stakeholder adds detailed information about the requirement, reasons to added, importance degree then share the report to other stakeholders. Accepting or rejecting the requirement is decided by project manager and team members. The team will calculate the budget and schedule change if the requirement is added. Then they will control the impact of requirement on product scope and quality. They will then analyze the risk factors that caused or changed by the requirement.

The Requirement Control Report is used when a requirement is failed to fulfilled or needed to be changed. Project manager add requirement information, reasons of failure, possible results of this requirement failure, and possible actions to take.

Schedule Control Report is used for informing stakeholders in an update on the schedule change or at the end of a milestone. If schedule is behind the plan, project manager add reasons to schedule change, new critical path, new schedule otherwise give information of current status of schedule.

Schedule Change Request is used when a team member wants to change schedule. Report includes reasons of change, possible outcomes and detailed description of change.

Budget Control Report is used for informing stakeholders if actual cost cannot follow budget plan or project is at the milestone. If it is the first case, project manager adds the detailed description of difference on budget plan and actual cost, reasons of it, possible outcome and actions that can be taken else adds the current status of budget.

Quality Control Report is used for informing stakeholders in any change on the quality plan.

These reports will be explained and shared with team members and sponsors in a meeting.

3.4 Risk Management Plan

First of all, as a team we identified risks and with the input, our Project Manager assessed the probability and impact of occurrence for each identified risk. We considered probability as;

- High – Greater than <70%> probability of occurrence
- Medium – Between <30%> and <70%> probability of occurrence

- Low – Below <30%> probability of occurrence

Risks	Category	Probability (%)	Impact
Misunderstanding of requirements	Product Size	30	High
Team is unfamiliar with the tool	Experience of Staff	70	High
Absence of any personnel	Staff	75	High
Design and coding deficiencies.	Process Definition	70	High
Change of a requirement by customer	Product Size	20	Low
Documentation does not serve its purpose	Process Definition	30	Medium
Incorrect and missing Work package definition	Process Definition	45	Medium
Less reuse than planned	Product Size	25	Low
Process demands not adequately planned in estimates	Product size	15	Low

Table 1- Identified Risks

There are four major risk which fall within RED zones since the probability and the impact of them are high. These risks are identified below. In order to mitigate the risks, we planned some alternatives.

Priority	Risk Scenarios	Response Strategy/Risk Alternatives	Monitoring	Assigned Person
1	Requirements are misunderstood	Strategy: Mitigate Alternatives: Schedule a meeting with customer and talk till everything is clear.	During the group meetings.	Kübra Nur Güzel

1	One of the members is sick or won't be attending to meetings or won't take any responsibility.	Strategy: Mitigate Alternatives: Other group members are over work. Or CarMarket can hire someone.	During the requirements phase	Arif Can Terzioğlu
2	Communication skill is not enough: In the code phase, the new requirements will be met.	Strategy: Mitigate Alternatives: Group will make a meeting with customers.	Middle of the coding phase.	Ömer Berk Uçar
2	The development tool is hard to use.	Strategy: Mitigate Alternatives: Before implementation CarMarket team will make mini projects with development tool.	During the implementation phase.	Pınar Bayata

Table 2 - Alternatives for High Impact Risks

3.5 Project Closure Plan

A final report including lessons learned and analysis of project objectives will be prepared, after the meeting between manager and team which will be organized to discuss the improvements of the project. CarMarket project will end in 31 December 2018. To get customer acceptance, a meeting will be organized between manager and customer. CarMarket team will provide the consultant who is our lecturer, Haluk Altunel, a copy of all the documents in electronic format. Team will archive all the project's artifacts such as source code, documents, project plans, user documentation and so forth.

After all of these processes are completed, team will make a demonstration of the project to the customer and sponsor. Supplier contracts and agreements will be closed. Project resources will be released to the business. Any outstanding activities or deliverables

will be listed. Team will complete a Post Implementation Review to identify the level of success of the project. An online User Manual will be provided.

4. Technical Process Plan

4.1 Process Model

The process model of the project is an incremental model. All work activities of the Work Breakdown Structure includes major milestones, project deliverables of each phase, reviews and schedules. This project will follow an incremental model for the following reasons:

- Core and urgent functionalities should be implemented first.
- Rapid delivery of the project can be possible even if the whole system is not developed completely.
- Some features that are not vital will be added after the essential parts with an incremental method.
- The system can be updated according to customer feedback.
- Development schedule can be divided into parts for the ease of feasibility.

4.2 Methods, Tools and Techniques

- Agile Methodology will be used for CarMarket.
- For the development of the system, MySQL (version 5.7) database will be used.
- The system should run on Windows Server 2003.
- PHP (version 5.5) will be used for backend development.
- HTML and CSS (latest versions) for the frontend development will be used.
- The interaction between frontend and backend will be easily manageable.
- The system should work on every web browser that supports HTML.
- Weekly reports and milestones will be created as electronic documents.
- For the diagrams, Visual Paradigm 14.0 is going to be used.
- Mockups of further user interfaces will be drawn with Balsamiq Mockups (version 3.5.7)

4.3 Infrastructure Plan

- There is no need for a physical facility or an office space for the project. Every member can work separately but the weekly scrum meetings should be done in a present environment which will be decided by the project manager.
- Daily scrum meetings can be held online for 15-20 minutes.
- Connection to the server system should be secure.
- Updates of each development will be pushed.
- Slack will be used for project specific discussions.
- Licences for software will be purchased by the company and which licence should be gotten will be decided by the manager. Costs for these operations are included in the project's budget.

4.4 Product Acceptance Plan

- Documents will be delivered on predefined dates.
- Documents will be prepared according to IEEE Standards.
- Each document will be updated due to feedbacks.
- Since Agile Methodology will be used, presentations will be held for certain milestones and get thoughts from customers.
- Acceptance, verification and validation of the work outcomes in each process are fundamental.
- The acceptance criterion is the customers' satisfaction with the project.
- After each phase, any feedback, suggestion or comments will be analyzed for the next phase. If it is considered a necessary implementation for those feedbacks will be started.

5. Supporting Plans

5.1 Configuration Management

Team will use Chef Configuration Management tool to keep track of artifacts such as source code, documentation, problems, changes requested and changes made. This tool can be used in concert with Git to monitor new releases. All of the team members are responsible to report, edit, trace the changes. Requested changes to software, hardware, documentation are reviewed before approved or denied since these changes will affect

security, performance, interfaces, completeness and documentation. Configuration Management Items for CarMarket are software and database components such as database components, computer programs which CarMarket uses, also operating system.

5.2 Quality Assurance Plan

Configuration Management methods will be used for quality assurance plan. In this phase, implementation of test cases will be evaluated, requirements will be checked. Since all of the team members are responsible for the changes made, all members should supervise the changes. Risk management plan should be used in order to solve risks created by the new changes. A new document containing the new data should be presented to project manager and according to this document, manager will approve or disapprove the quality of the project.

5.3 Process Improvement Plan

First step for the process improvement plan is to selecting a process to be improved such as implementation. A team will be formed. If it is implementation process that needs improvement, then a team of developers will be formed. After defining and collecting data about the process, the process will be simplified with getting rid of redundant or unnecessary activities. Baseline comparison will be made with the data collected and problems with the existing process will be covered. Configuration management plan will be used to implement the change in the process. Optimized process will be tested and the team formed will assess whether the changed process has actually been improved.

6. Additional Plans

For assuring safety, privacy, and security requirements are met we will use third party services. Cloudflare service will be used to guarantee availability all the time and to mitigate attacks as much as possible. 3D secure will be used for privacy and security at payment processes.

In future the system will be expanded to provide sale on other categories like motorcycles, scooters, trucks etc. All categories will be added as submarket into the system and users can explore all market from the same webpage.

Appendixes

Task Name	Duration	Start	Finish	Predecessors	Q1 Q2 Q3 Q4											
					Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1.0. Car Market	197d	01/06/19	10/07/19													
2.0. Project Management	31d	01/06/19	02/15/19				Kübra Nur Güzel									
2.1. Planning	5d	01/06/19	01/10/19													
2.2. Requirement Management	10d	01/07/19	01/18/19	3SS												
2.3. Time and cost Management	10d	01/07/19	01/18/19	4SS												
2.5. Quality Management	10d	01/21/19	02/01/19	5												
2.4. Risk Management	10d	01/21/19	02/01/19	6SS												
2.6. Resource Management	10d	01/21/19	02/01/19	7SS												
2.7. Documentation	10d	02/04/19	02/15/19	8												
3.0. Trainings	5d	01/21/19	01/25/19	8SS			Instructors									
3.1. Defining development Tools	2d	01/21/19	01/22/19	8SS												
3.2. Deciding familiarities of Tool	3d	01/21/19	01/23/19	8SS												
3.3. Team members' orientation	5d	01/21/19	01/25/19	8SS												
4.0. System Engineering	16d	02/18/19	03/11/19	2			Arif Can Terzioğlu									
4.1. Planning of technical background	6d	02/18/19	02/25/19													
4.2. Defining system requirements	8d	02/18/19	02/27/19													
4.3. Design high level system architecture	10d	02/26/19	03/11/19	22												
4.4. Design low level system architecture	10d	02/26/19	03/11/19	22												
5.0. Software Engineering	166d	02/18/19	10/07/19													
5.1. Requirement Elicitation	10d	02/18/19	03/01/19	2			Kübra Nur Güzel									
5.1.1. Software requirement specification	10d	02/18/19	03/01/19													
5.1.2. Customer requirements	6d	02/18/19	02/25/19													
5.1.3. Development team requirements	4d	02/26/19	03/01/19	22												
5.2. Software Analysis	3d	03/04/19	03/06/19	23			Kübra Nur Güzel									
5.2.1. Assigning task to team	3d	03/04/19	03/06/19	23												
5.3. Software Design	18d	02/18/19	03/13/19													
5.3.1. GUI Design	15d	02/18/19	03/08/19	21SS			Pınar Bayata									
5.3.1.1. User Interface Design	15d	02/18/19	03/08/19	21SS												
5.3.2. System Design	12d	02/26/19	03/13/19	17SS			Arif Can Terzioğlu									
5.3.2.1. Unit Design	8d	02/26/19	03/07/19	17SS												
5.3.2.2. Component Design	11d	02/26/19	03/12/19	17SS												
5.3.2.3. System Design	12d	02/26/19	03/13/19	17SS												
5.3.3. Database Design	12d	02/26/19	03/13/19	18SS			Database Analyst									
5.3.3.1. Creating Tables	6d	02/26/19	03/05/19													
5.3.3.2. Constraints	6d	03/06/19	03/13/19	34												
5.3.3.3. Creating relations	6d	03/06/19	03/13/19	34												
5.4. Software Implementation	130d	03/12/19	09/09/19	29SS + 10d												
5.4.1. Software Unit coding	100d	03/12/19	07/29/19	30,33												
5.4.1.1. Back-end Services	82d	03/12/19	07/03/19													
5.4.1.1.1. API Integration	37d	03/12/19	05/01/19													
5.4.1.1.2. Server Requests / Functions	45d	05/02/19	07/03/19	40												
5.4.1.2. Login System	15d	05/02/19	05/22/19	41SS												
5.4.1.3. Search System	18d	07/04/19	07/29/19	41												
5.4.1.3.1. Filtering	18d	07/04/19	07/29/19													
5.4.1.4. Recommendation System	25d	05/02/19	06/05/19	41SS												
5.4.1.5. Posting Mechanism / Page	15d	05/02/19	05/22/19	41SS												
5.4.1.6. Bid System	20d	05/02/19	05/29/19	41SS												
5.4.1.7. Payment	30d	03/12/19	04/22/19	40SS												
5.4.1.9. Driver /Delivery Tracking System	20d	03/12/19	04/08/19	54,41SS												
5.4.1.8. Message System / Page	20d	03/12/19	04/08/19	41,52SS												
5.4.1.10. Main Page	30d	06/06/19	07/17/19	45												
5.4.1.11. Advertisement Page	15d	05/02/19	05/22/19	41SS												
5.4.1.12. Users' Posts / Favorite Posts Page	15d	05/02/19	05/22/19	41SS												
5.4.1.13. Purchased Items / Delivery Page	15d	05/02/19	05/22/19	41SS												
5.4.1.14. Notifications Section	15d	03/12/19	04/01/19	40SS												
5.4.2. Unit debugging	20d	08/13/19	09/09/19	59SS + 10d												
5.5. Testing	50d	07/30/19	10/07/19													
5.5.1. Testing Plan	10d	07/30/19	08/12/19	38												
5.5.2. Unit Testing	20d	07/30/19	08/26/19	38												
5.5.3. Component Testing	15d	08/27/19	09/16/19	59												
5.5.4. System Testing	15d	09/17/19	10/07/19	60												
5.5.5. Database transaction testing	15d	09/17/19	10/07/19	61FF												
6.0. Deployment & Release Activities	42d	07/30/19	09/25/19	57SS												
6.1. Deployment Management	10d	07/30/19	08/12/19													
6.2. Configuration Management	7d	08/13/19	08/21/19	64												
6.3. Change Management	8d	08/22/19	09/02/19	65												
6.4. Verification & Validation	7d	09/03/19	09/11/19	66												
6.5. Acceptance Testing	10d	09/12/19	09/25/19	67												
7.0. Supporting Activities	10d	09/12/19	09/25/19													
7.1. Quality Assurance	10d	09/12/19	09/25/19	68SS												
8.0. Closure	5d	09/26/19	10/02/19	63												
8.1. User Manual	3d	09/26/19	09/30/19	63												
8.2. End user training	2d	10/01/19	10/02/19	72												