GPIG – Initial Report

Department of Computer Science, University of York

Group D

Kieran McHugh (KM)

Liam Wellacott (LW)

Andrei Zisu (AZ)

Lloyd Still (LS)

Oliver Lea (OL)

Mark Woosey (MW)

Paulius Kazakas (PK)

Submission on 11th May 2017

Contents

1 Introduction

Unsustainable growth in the cargo shipping industry, paired with a lack of innovation, puts incredible pressure on the environment and threatens to destabilise the economy. In this report, we illustrate how we intend to leverage collaborative autonomy to develop a more sustainable approach. We describe a possible solution involving the use of 'off-shore ports' along with small, autonomous shuttle boats which serve to provide a link with traditional coastal ports.

We begin with a discussion of the challenges facing the industry, and we make some informed assumptions about the capabilities of the technology at our disposal in the year 2030. We then describe our proposed solution, outlining key benefits, and explaining how we will justify our approach through simulation. We also provide a brief summary of our team structure, the key processes we will follow, and our customer communication strategy. We conclude with a register of significant project risks.

2 Problem Analysis

3   System Description

4 Advantages and Benefits

5 System Prototype

6   Team Structure and Roles

We elected at an early stage to adopt a well-defined team structure by assigning a logistical role to each member. Members conduct their logistical role in addition to any writing or engineering tasks assigned to them. The assignment of members to roles was a group exercise. We took into account individual strengths and areas of previous experience.

Role	Member	Primary Responsibilities	Previous Experience
Chair	KM	Compile the agenda for each meeting; run each meeting; maintain the risk register; distribution of writing tasks.	Kieran acted as an academic representative and Chair of the departmental Staff Student forum for two years.
Secretary	LW	Organise each meeting; take minutes for meetings; assign actions; distribution of writing tasks; report submissions.	During his placement, Liam organised and ran several client meetings with stakeholders for his project.
Product Owner	AZ	Perform code quality reviews; review and merge pull requests; resolve merge conflicts; code repository management.	Andrei is an active volunteer for large open source projects, conducting code reviews and synchronising contributors.
Scrum Master	LS	Assign 'story points' and risk values to each task; collect productivity and code quality metrics; coordinate 'stand up' and 'scrum' meetings.	During placement, Lloyd has participated in and lead agile meetings. He also managed metrics collection for his placement project.
Lead Architect	MW	Decide the core technologies, languages and libraries to use; create project code structure; identify control/data flows, model structures etc.	On placement, Mark worked with a very broad range of platforms and languages. He has studied system architecture and model driven engineering modules.
Lead Engineer	OL	Oversee development activities; assign programming tasks to members; ensure compatibility of different parts of the solution; coordinate pair programming.	Oliver has lead software projects across multiple international companies and has a strong understanding of software design principles.
Customer Relations	PK	Acting as the single point of contact with the customer; maintaining records of discussions with customer; keeping customer informed of progress.	Having dealt with internal customers on a daily basis during placement, Paulius has experience addressing, investigating and solving customer problems.

7 Key Processes

8 Communication Strategy

9   Risk Register

We identified the following ongoing risks, assigning a mitigation strategy to each, as well as a relevant owner based on the roles described in Section 6.

ID	Description	Likelihood	Impact	Owner	Mitigation
1	Team members are unable to complete writing tasks in a timely fashion due to illness or preoccupation with other work.	Low	High	KM/LW	
2	Team members struggle to complete assigned programming tasks in time due to lack of experience with the chosen technologies.	Medium	High	OL	
3	The chosen languages, frameworks, or libraries are too difficult to learn, have compatibility issues, or do not have the expected capabilities.	Medium	High	MW	
4	Progress on programming tasks is delayed due to poor code quality, duplication of effort, or conflicting code commits by team members.	Medium	Medium	AZ	
5	The customer is not available to give comments on proposed changes or additions to the system specification.	Medium	Low	PK	
6	Incomplete metrics due to members not completing the daily online 'stand-up', forgetting to log their activity, or not maintaining their action tickets.	High	Low	LS	
7	Loss of writing or code due to failure of team members' hardware and lack of regular backup/commit.	Low	High	KM/OL	
8	Team members are not aware of assigned tasks, or complete the wrong tasks, due to not attending meetings, or being late to meetings.	High	Medium	LW	
9	Difficulty completing programming tasks on time due to underestimation of work involved or too broad a scope.	Medium	High	MW/OL	
10	Delays in creating the prototype due to the identification of a major design flaw in the system specification.	Low	High	KM/LW	

10 Conclusion

References