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 'offshore smart 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

With more than 90% of all international trade conducted by sea, the shipping industry plays a crucial role in the modern interconnected world [1]. Over the past 50 years, the size of the world shipping fleet has increased from 36,000 to over 100,000 [1]. In order to maximise efficiency of transportation [2], the capacities of these ships have increased substantially. In 1956, for instance, a typical container ship had a capacity of 101 Twenty-foot Equivalent Units (TEUs) [3]. By comparison, ships operating in 2015 have capacities of up to 20,000 TEUs [2].

The rapid expansion in both fleet size and freight capacity raises many questions about the sustainability of the shipping industry. In 2016, freight rates (the wholesale price of shipping) fell to the lowest levels since the 2008 global recession. This caused Hanjin, a major operator in the shipping industry, to enter administration [4]. We argue that, if existing companies wish to meet future demand and survive further economic turmoil [5], the industry must adapt.

Since the growth of seaborne trade shows little sign of slowing, we can expect to see further increases in both the number of ships and their capacities. This necessitates the development of coastal infrastructure to accommodate larger vessels, putting extra pressure on already taxed resources [3]. At the same time, there will be increasing pressure from political bodies, such as the EU, to reduce the impact the industry is having on the environment [6]. This is forcing the industry to explore 'greener' approaches, perhaps at the cost of more economical alternatives.

Collaboration and autonomy in shipping systems have the potential to mitigate these problems, and many techniques are in development today. For instance, Rolls-Royce plan to bring autonomous container ships into service by 2020 [7]. These ships will combine data from satellites, weather reports, and an array of LIDAR and infrared sensors to enable completely unmanned navigation. It is therefore reasonable to assume that by 2030, these techniques will be commonplace and can be incorporate into our solution.

Autonomous ships generate a wealth of data, which can be leveraged by other ships and ports to make more informed decisions [5]. The automotive industry is already exploring ways in which the data generated by autonomous vehicles can be applied to solve pressing problems. One proposed system collects broadcasted information from other vehicles, such as speed and position, to create a model of the environment and provide the driver with an early warning system for potential hazards [8]. With development of these systems already in progress, we believe that the shipping industry must consider investment into collaborative and autonomous technologies.

3 System Description

To mitigate some of the issues discussed in Section x, we propose a new system taking inspiration from the decentralised nature of land-based courier networks. Parcels in transit over land typically pass through several large distribution centres, taking an indirect path to their destination. While this means each individual parcel travels further, it greatly improves the efficiency of the system as a whole. Our system will leverage collaborative autonomy in the form of a fleet of autonomous shuttle ships. These ships will connect a series of 'offshore smart ports' to existing coastal infrastructure.

¡¿

Each offshore smart port will act as a regional hub, accepting cargo from existing intercontinental freight ships. This cargo is subsequently forwarded to several coastal ports. The offshore ports are stationary and fixed to the seabed, in much the same way as oil extraction platforms. Aboard each offshore port is an automated container storage and transit system inspired by automated parking facilities [9]. This system will enable simple storage and retrieval of individual containers through the avoidance of stacking. When each container is ready to leave the offshore port, it is automatically routed to a crane for transfer to a docked vessel. The offshore port will be powered with an onsite nuclear reactor. We assume that it will be both technically feasible and within nuclear regulations to deploy such a reactor to our offshore ports in 2030.

The offshore smart ports will be serviced by a fleet of small autonomous shuttle ships, each capable of transporting a varying number of freight containers between the coastal and offshore ports. The fleet will employ swarm intelligence mechanisms to self-organise. This coordination will help individual ships plan their paths around bad weather and other obstacles, and will also allow the system to respond dynamically to demand at different coastal ports. When goods at a coastal port are ready for dispatch, the shuttle ships will collectively decide an optimal assignment strategy based on their proximity to the collection point. On arrival at the offshore port, the vessel communicates with the storage system to provide information regarding the final destination of the carried goods. Control of the vessel is then delegated to a control system on the offshore port, which is responsible for positioning of the vessel so that the cargo can be offloaded. The shuttle ships will be powered primarily by an on board battery, which is charged at port when necessary. Additionally, on-board solar panels will be used as a means of emergency power generation. We assume that battery technology has developed sufficiently to support the powering of these autonomous vessels for a journey between ports. We also assume that these batteries will be charged using 'green' energy sources.

¡¿

Although system components will be fully autonomous, control of the system is transferable to remote human operators in exceptional circumstances such as malfunction or disaster in order to mitigate potential adverse consequences of these events. The offshore port and autonomous fleet will initially function at a single location. However we envision that a network of these systems will be deployed at key locations around the world.

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

The team will follow agile practices by conducting regular 'scrum' meetings. The aim of these meetings is to gather system requirements, and break down the development of large, complex features into a series of actionable tasks. These tasks are each assigned a number of 'story points' depending on their complexity, added value, and thus the amount of time they will take to complete.

The use of kanban boards on GitHub to track the status of each task allows for the effective management of feature development, defect tracking and writing tasks. The tasks will be distributed among team members using a pair programming scheme, with the aim of ensuring the correctness and quality of initial code [10]. The persons responsible for each task will maintain and update the status of their tickets on the kanban board.

Progress will be presented to the team in the form of daily 'stand ups'. The purpose of the stand ups is to ensure the team is focused, and consistently progressing towards its goals. Stand ups will take place online using Slack, where each team member will report the following:

- What progress they have made since the previous stand up;
- What tasks and goals they are currently working towards;
- What obstacles they anticipate facing, and what methods they will use to overcome them;
- What progress they aim to have achieved before the next stand up.

We will conduct unit tests in order to assess the functionality of each component. In addition to continual code review within each pair, code will be reviewed before being merged to the main branch. This is handled by the designated product owner, to ensure compatibility between functional components.

8 Metrics

In an agile development setting, measuring progress is often difficult. Hartmann and Dymond [11] identify 11 properties of effective agile measurements. It is important for metrics to be meaningful, and focus on highlighting delivered value as opposed to output quantity. This ensures that the metrics accurately represent the team's progress, and the quality of the product. In view of these properties, we opt to measure the following key performance indicators:

Story Points Completed per Day – represents the velocity of progress. More complex tasks are assigned more story points. Therefore, the number of story points completed per day acts as an accurate measure of how much work is actually being completed – more so than quantitative output metrics such as lines of code [12]. Measuring on a per day basis allows for trends to quickly be established, despite the short project duration.

Estimated vs. Actual Time – represents the deviation from our initial effort estimates, plotted on a graph on a daily basis. Deviation from the estimated time has been shown to be an effective performance measure, and will act as an indicator that team efforts need distributing more effectively [13].

Number and Severity of Faults per Day – represents the maturity and correctness of the system. By logging number of defects alone, we cannot obtain a representative measure of maturity. This is because a fault could be a simple one line code change, or it might require significant modifications to functionality. We therefore opt to log the severity of faults on a scale of Minor, Major and Critical – with the aim of observing a decrease in severity of faults over time.

Team Satisfaction – represents the extent to which team members are satisfied with both their own, and team progress. Measured on a linear scale that ranges from 1 (very dissatisfied) to 5 (very satisfied). Team morale has been shown to provide an important indication as to how successful the final outcome will be [14].

The collection of these metrics will be aided by the use of kanban boards and a Gantt chart, allowing us to keep track of development time, story points, number of defects and fault severity. Soft metrics such as team satisfaction will be collected via an automated process in Slack.

9 Communication Strategy

10 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	

11 Conclusion

References

[1] International Maritime Organisation, "International shipping facts and figures – information resources on trade, safety, security, environment," 2012. [Online]. Available: http://www.imo.org/en/KnowledgeCentre/ShipsAndShippingFactsAndFigures/TheRoleandImportanceofInternationalShipping/Documents/International%20Shipping%20-%20Facts%20and%20Figures.pdf

[2] ABB. (2016) Maritime cargo vessels – is bigger better? [Online]. Available: http://new.abb.com/turbocharging/maritime-cargo-vessels---is-bigger-better

[3] FT. (2016) Shipping's size obsession could be ending, study finds. [Online]. Available: https://www.ft.com/content/255d9394-e47a-11e5-a09b-1f8b0d268c39

[4] A. Tovey. (2016) Global shipping crisis deepens as cargo line nyk takes multi-billion yen hit. [Online]. Available: http://www.telegraph.co.uk/business/2016/10/07/global-shipping-crisis-deepens-as-cargo-line-nyk-takes-multi-bil/

[5] IHS, "Five trends shaping the global maritime industry," 2015. [Online]. Available: https://www.ihs.com/pdf/Global-Trends-Impacting-the-Maritime-Industry_235788110915583632.pdf

[6] ESPAS, "Global trends to 2030: Can the eu meet the challenges ahead?" 2015. [Online]. Available: http://ec.europa.eu/epsc/sites/epsc/files/espas-report-2015.pdf

[7] O. Mitchell. (2017) Autonomous boats by 2020. [Online]. Available: https://hackernoon.com/autonomous-boats-by-2020-7929a0d9010f

[8] W. Knight. (2015) Car-to-car communication. [Online]. Available: https://www.technologyreview.com/s/534981/car-to-car-communication/

[9] A. Mathijssen and A. J. Pretorius, *Verified Design of an Automated Parking Garage*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 165–180.

[10] J. E. Hannay, T. Dybå, E. Arisholm, and D. I. Sjøberg, "The effectiveness of pair programming: A meta-analysis," *Information and Software Technology*, vol. 51, no. 7, pp. 1110–1122, 2009.

[11] D. Hartmann and R. Dymond, "Appropriate agile measurement: using metrics and diagnostics to deliver business value," in *Agile Conference, 2006*. IEEE, 2006, pp. 6–pp.

[12] C. W. Davis, *Agile metrics in action*. Manning Publications,, 2015.

[13] D. R. Greening, "Agile enterprise metrics," in *48th Hawaii International Conference on System Sciences (HICSS)*. IEEE, 2015, pp. 5038–5044.

[14] ProWareNess. Agile metrics: Let the numbers tell the tale. [Online]. Available: http://www.scrum.nl/media/Agile_Metrics/$FILE/whitepaper_agile_metrics.pdf