Feasibility Evidence Description (FED)

We Are Trojans (WAT) Network

Team01

Team members	Roles
Eirik Skogstad	Project Manager, Life Cycle Planner
Min Li	Feasibility Analyst, Operational Concept Engineer
Pittawat Pamornchaisirikij	NDI/NCS Acquirer & Evaluator, Tester
Saloni Priya	Requirements Engineer, UML Modeler
Suleyman Erten	Operational Concept Engineer, Requirement Engineer
Kamonphop Srisopha	Prototyper, UML modeler
Ameer Elkordy	IIV&V, Quality Focal Point
Punyawee Pakdiying	System Architect, Feasibility Analyst

11/28/2014

Version History

Date	Author	Version	Changes made	Rationale
09/28/14	ML, PP	1.0	• Create initial a FED document from a template, updating the risk assessment section.	• For use VCP package submission of the project.
10/11/14	ML, PP	1.5	• Finish all from section 1 to 5	• For use VCP package submission of the project.
10/19/14	ML, PP	2.0	• Updated all section • Make consistent with ARB and FCR presentation • Some NDI was evaluated.	• Use in next phase (Foundation phase) • To be consistent with ABR presentation
10/29/14	PP	2.1	• Added more NDI/NCS analysis and evaluation • Update the risk list	• Consider more NDI/NCS to gain more information and reduce risk in development phase
11/16/14	ML, PP	2.5	• Change to Architected Agile Template • Update the risk list • Update LOS • Added ROI analysis • Fixed mistake from version 2.1	• Update the risk list related to progress report file. • Make LOS to be consistent with OCD file

Table of Contents

Feasibility Evidence Description (FED) i

Version History ii

Table of Contents iii

Table of Tables iv

Table of Figures v

1. Introduction 1

1.1 Purpose of the FED Document 1

1.2 Status of the FED Document 1

2. Business Case Analysis 2

2.1 Cost Analysis 3

2.2 Benefit Analysis 3

2.3 ROI Analysis 4

3. Architecture Feasibility 5

3.1 Level of Service Feasibility 5

3.2 Capability Feasibility 5

3.3 Evolutionary Feasibility 6

4. Process Feasibility 8

5. Risk Assessment 11

6. NDI/NCS Interoperability Analysis 13

6.1 Introduction 13

6.2 Evaluation Summary 13

Table of Tables

Table 1: Personnel Costs 3
Table 2: Hardware and Software Costs 3
Table 3: Benefits of xxx System 4
Table 4: ROI Analysis 4
Table 5: Level of Service Feasibility 5
Table 6: Capability Requirements and Their Feasibility Evidence 5
Table 7: Evolutionary Requirements and Their Feasibility Evidence 7
Table 8: Rationales for Selecting Architected Agile Model 8
Table 9: Risk Assessment 11
Table 10: NDI Products Listing 13
Table 11: NDI Evaluation 14

Table of Figures

Figure 1: ROI Analysis Graph *4*

1. Introduction

1.1 Purpose of the FED Document

This document reports our analysis about the feasibility evidence of the We Are Trojans (WAT) Network project. We use risk assessment to identify and come up with a way to mitigate those risks. We will analyze NDI items and evaluate the risk if whether they fit our project.

1.2 Status of the FED Document

This is version 2.2 of the FED document.

- Template of this document has been changed to Architected Agile template because we figured that even we use CMS, we also have to modified a lot of CMS code.
- ROI Analysis that we get the information from our clients was added.
- Possible of Users may prefer existing systems with similar features Risk and User may not accept to use the system even if all the specification are met by the system Risk have been lower because the result of the survey we conducted.
- Mistakes from the past version was fixed.

2. Business Case Analysis

<table>
<tr><td colspan="4">Assumptions
• USC students need a central platform to connect, share, and like information with each other
• Reward point system will work as the important incentives for users to join the network</td></tr>
<tr><th>Stakeholders</th><th>Initiatives</th><th>Value Propositions</th><th>Beneficiaries</th></tr>
<tr><td>• Developers
• Maintainers
• Clients
• Gift/Book stores</td><td>• Develop the system
• Monitor the system
• Advertise the system to USC community
• Partner with schools
• Negotiate deals with on-campus bookstore/gift store
• Join the system</td><td>• Increase camaraderie between Trojans
• One-stop shop to answer any USC related queries
• Increase communications between students across schools</td><td>• USC students
• USC alumni
• USC faculties</td></tr>
<tr><td colspan="2">Cost
• Development costs
• Maintenance costs
• Advertising/Marketing costs
• Web server, Web hosting, Domain name</td><td colspan="2">Benefits
• The number of active users in “WAT” network increases.</td></tr>
</table>

2.1 Cost Analysis

2.1.1 Personnel Costs

Table 1: Personnel Costs

Activities	Time Spent (Hours)
Development Period (24 weeks)	
Valuation and Foundations Phases: Time Invested (CSCI577a, 12 weeks)	
Client and team: Meeting via email, phone, and other channels [3 hrs/week * 12 weeks * 2 people]	72
winwin sessions [2 winwin session * 1 hours * 2 people]	4
Architecture review boards [1.5 hours * 2 session * 2 people]	6
Development and Operation Phases: Time Invested (CSCI577b, 12 weeks)	
Client: Meeting via email, phone, and other channels [3 hrs/week * 12 weeks * 2 people]	72
Architecture Review Boards and Core Capability Drive-through session [1.5 hours * 2 session * 2 people]	6
Deployment of system in operation phase and training - Installation & Deployment [5 hrs * 2 times *2 people] - Training & Support [5 hrs * 2 times * 2 people]	40
Total	200
Maintenance Period (1 year)	
Maintenance	210
Total	**410**

2.1.2 Hardware and Software Costs

Table 2: Hardware and Software Costs

Type	Cost	Rationale
-	-	-

*There are no Hardware and Software costs because we will select only free software and webhosting.

2.2 Benefit Analysis

Table 3: Benefits of xxx System

Current activities & resources used	% Reduce	Time Saved (Hours/Year)
USC Tech Support		
Student Support (8hrs * 5days = 40 hours)	25%	520
Faculty Office Hours		
Faculty Support (3hrs/week = 3 hours)	33%	52
Total		**572**

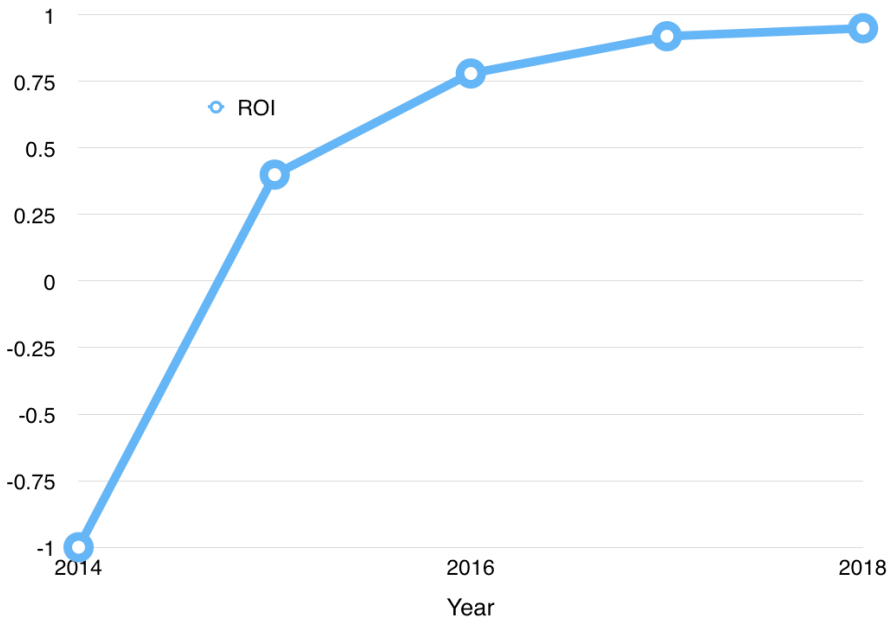
2.3 ROI Analysis

Table 4: ROI Analysis

Year	Cost	Benefit (Effort Saved)	Cumulative Cost	Cumulative Benefit	ROI
2014	200	0	200	0	-1
2015	210	572	410	572	0.40
2016	231	572	641	1144	0.78
2017	254.1	572	895.1	1716	0.92
2018	279.5	572	1174.6	2288	0.95

*Assuming 10% per year increase in cost

Figure 1: ROI Analysis Graph



3. Architecture Feasibility

3.1 Level of Service Feasibility

Table 5: Level of Service Feasibility

Level of Service Requirement	Product Satisfaction
LOS-1: The system shall be user-friendly and intuitive.	Product Strategies: Apache, MySQL, Laravel, JQuery, Bootstrap, Google survey
	Process Strategies: Prototyping, Survey
	Analysis: We will create prototype that have dynamic user interface and then conduct a survey. Our system will be user-friendly and intuitive if the result show that more than 80% of user agree that our system is easy to use and intuitive.
LOS-2: The system shall render correctly on mobile platform	Product Strategies: Apache, Laravel, JQuery, Bootstrap
	Process Strategies: Prototyping, Analysis and evaluate NDI, NCS
	Analysis: We will use Bootstrap that have dynamic rendering feature which will render differently on mobile platform.

3.2 Capability Feasibility

Table 6: Capability Requirements and Their Feasibility Evidence

Capability Requirement	Product Satisfaction
CR-1: Q&A Forum	Software/Technology used: Laravel, MySQL, JQuery
	Feasibility Evidence: phphub.org and www.tasty.lk have similar capability which can prove CR-1 is feasible. We also identified that Laravel API can connect, edit, update, delete, and retrieve information from database. JQeury can send an event to our server.
	Referred use case diagram: Figure 3 in SSAD file.
CR-2: WAT Point System	Software/Technology used: Laravel, MySQL
	Feasibility Evidence: This capability feasibility has shown in our Prototype that we create algorithm to calculate WAT points.
	Referred use case diagram: Figure 3 in SSAD file.

CR-3: Notification System	Software/Technology used: Laravel, MySQL, Bootstrap, JQuery, PusherNotifier.js
	Feasibility Evidence: Laravel can get specific information from MySQL. PusherNotifier.js can push notifications from our server. We identified from caniuse.com that web sockets (PusherNotifier.js) can use in IE, Firefox, Chrome, Safari many popular web browser.
	Referred use case diagram: Figure 3 in SSAD file.
CR-4: Profile	Software/Technology used: This is the same as CR-1
	Feasibility Evidence: This is the same as CR-1. Because in essence, This capability need to connect and retrieve and update user profile data from database to show it on the user profile page.
	Referred use case diagram: Figure 3 in SSAD file.
CR-5: Leaderboard	Software/Technology used: Laravel, MySQL, Bootstrap, JQuery
	Feasibility Evidence: This is almost the same as CR-1. Except we have to sort data by users' Semester point and show on the page which can be done using Eloquent ORM that included in Laravel to sort the data.
	Referred use case diagram: Figure 3 in SSAD file.
CR-6:Redemption	Software/Technology used: Laravel, MySQL, Bootstrap, JQuery
	Feasibility Evidence: Evidence: www.piccologifts.co.uk and superbalist.com is examples of web that is built by Laravel that have similar capability which can prove CR-5 is feasible. Similar to CR-1, we already test related Laravel API and JQuery function that ensure this is feasible.
	Referred use case diagram: Figure 3 in SSAD file.
CR-7: Event System	Software/Technology used: This is the same as CR-1
	Feasibility Evidence: This is the same as CR-1 because an event is a special thread that created by a maintainer.
	Referred use case diagram: Figure 3 in SSAD file.

3.3 Evolutionary Feasibility

*We have no evolutionary feasibility because we have to wait for our client to talk to USC that we can integrate with USC system first.

Table 7: Evolutionary Requirements and Their Feasibility Evidence

Evolutionary Requirement	Rationale
-	Software/Technology used: -
	Feasibility Evidence: -
	Referred use case diagram: -

4. Process Feasibility

Decision Criteria Rating Scale; Very Low; Low; Medium; High; Very High
Importance Rating Scale: Low; Medium; High

Table 8: Rationales for Selecting Architected Agile Model

3Criteria	Importance	Project Status	Rationales
30 % of NDI/NCS features	Low	Low	We almost implement every feature because core feature (WAT points) in our system is unique.
Single NDI/NCS	Low	Low	We use more than one NDI. NDI that we are using just provide API to help us developing some specific feature. We can not find only single NDI that have all of the feature we want.
Unique/ inflexible business process	Low	Low	The business aspects of the project are very flexible. Because our requirement is flexible. There is no constrains in our project.
Need control over upgrade / maintenance	High	High	The project has to be upgraded in future after the client negotiating with the USC.
Rapid deployment	Low	Very low	Currently we are just building a dummy system. The system initially will not be deployed.
Critical on compatibility	Low	Very low	The system has no compatibility issue. Becaue we will built the system and then look for a web hosting that is compatible with our system. We also have no legacy system to concern with compatibility.

Internet connection independence	Low	Low	Internet connection is important, as the application developed is a web-based application.
Need high level of services / performance	Medium	Low	High level of services and performance is important. Because this is user-driven business. If our service is not good then this system will be fail.
Need high security	Medium	Medium	The system will be used only by USC students. That mean the size of information loss is not very critical like some information such as credit card.
Asynchronous communication	Medium	Medium	The system requires asynchronous communication to communicate with the web hosting.
Be accessed from anywhere	High	High	The system is an online community.
Critical on mass schedule constraints	Low	Very low	No, the system is not critical on mass schedule constraints. Because there is no win-condition about this.
Lack of personnel capability	Low	Very low	The group consists of highly competent graduate software engineers and because We Are Trojans!
Require little upfront costs	High	High	The budget for our project is $0, as per our client specifications.
Require low total cost of ownership	Medium	Medium	We requires no cost of ownership because we will use only open source software and free service.

Not-so-powerful local machines	High	High	We have minimal cost and we also have no infrastructure right now. We will be using free left over 8 year old laptops.

5. Risk Assessment

Table 9: Risk Assessment

Risks	Risk Exposure			Risk Mitigations
	Potential Magnitude	Probability Loss	Risk Exposure	
User may prefer existing systems with similar features because our system is new and they do not know about our system enough.	6	9	54	Make the WAT points system as an incentive to attract users over competitors. Advertize our system to USC and users. Create surveys and evaluate users' responses.
User Risk: users may not accept to use the system even if all the specification are met by the system	7	10	70	- We conduct the survey to acquire users' feedback regarding the project. - We will create the user interface mock up to acquire feedback from the clients
Lack of involvement by success-critical-stakeholders: Do not understand clearly enough the success-critical stakeholder positions	5	10	50	Potential users will be involved via conducting a survey and by letting them test prototypes of the user interface. Clients are involved by verifying that our work fits their expectations and by holding weekly meetings.
May not have sufficient time to create comprehensive prototypes for important parts of the system because we have many prototypes to create and almost all of them require	8	6	48	Prioritize the prototyping. Use minimum effort for each prototype to gain enough information to see the feature is feasible.

a lot of time.				
Undefined plan and Requirements: - The scope of the event functionality is not finalized. - The requirement about the UI of the system is not finalized. - The point system is still not verified by potential users - The scope of the search functionality is not certain Requirement have a tendency to change with project development.	5	9	45	- Consult with the clients how event functionality works - Prototyping will help us know more about point systems, search functionality, and UI - Get the feedback of surveys for UI

6. NDI/NCS Interoperability Analysis

6.1 Introduction

<< Identify the Non-Developmental Item (NDI) and Net-Centric Services (NCS) including open source software or libraries that you are using/ plan to use in your project and analyze their interoperability. >>

6.1.1 COTS / GOTS / ROTS / Open Source / NCS

Table 10: NDI Products Listing

NDI/NCS Products	Purposes
Laravel	PHP Framework, It provide general API and tool to implement PHP web application.
MySQL	DMBS, To manage database
Apache	Web server
JQuery	Javascript Library, which provide simple and easy-to-use javascript API.
Bootstrap	Front-end Framework that contains HTML and CSS based design templates

6.1.2 Connectors

<< Identify the connector, for example

- "In this project, we use PHP/MySQL Connector to enable the PHP web application to retrieve and query data from the database". >>

6.1.3 Legacy System

<< Identify the connector, for example

- "In this project, the development system has to be able to interoperate and works well with "BusinessWorks" version 5.2, which is a software system that the client is currently using." >>

6.2 Evaluation Summary

<< Summarize the final selection of your interoperable NDI/NCS, its usage and its comment. Example can be found in ICSM EPG> Task: Analyze NDI Interoperability for NDI / NCS project. >>

Table 11: NDI Evaluation

NDI	Usages	Comments