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

12/07/14

Version History

Date	Author	Version	Changes made	Rationale
09/28/14	ML, PP	1.0	• Created initial FED document from ICSM template, updated the risk assessment section.	• For draft VCP package submission.
10/11/14	ML, PP	1.5	• Finished all from section 1 to 5	• For VCP package submission.
10/19/14	ML, PP	2.0	• Updated all sections based on feedback. • Make consistent with ARB and FCR presentation • Updated NDI evaluations.	• Use in next phase (Foundation phase). • To be consistent with ABR presentation.
10/29/14	PP	2.1	• Added more NDI/NCS analysis and evaluations. • Updated the project risks.	• 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. • Updated the risk list. • Update LOS. • Added ROI analysis. • Fixed mistakes from version 2.1	• Update the risk list related to progress report file. • Make LOS to be consistent with OCD file.
12/7/14	ML, PP, ES, TS	3.0	• Change ROI analysis. • Fixed mistake from version 2.5 • Final review before submission.	• Changes in ROI calculations. • For DCP package submission.

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 4

2.3 ROI Analysis 5

3. Architecture Feasibility 6

3.1 Level of Service Feasibility 6

3.2 Capability Feasibility 6

3.3 Evolutionary Feasibility 8

4. Process Feasibility 9

5. Risk Assessment 12

6. NDI/NCS Interoperability Analysis 14

6.1 Introduction 14

6.2 Evaluation Summary ..... Error! Bookmark not defined.

Table of Tables

Table 1: Personnel Costs *3*
Table 2: Maintenance Cost per Year (hours) *4*
Table 3: ROI Analysis *5*
Table 4: Level of Service Feasibility *6*
Table 5: Capability Requirements and Their Feasibility Evidence *6*
Table 6: Evolutionary Requirements and Their Feasibility Evidence ..... ***Error! Bookmark not defined.***
Table 7: Rationales for Selecting Architected Agile Model *9*
Table 8: Risk Assessment *12*
Table 9: NDI Products Listing ..... ***Error! Bookmark not defined.***
Table 10: NDI Evaluation ..... ***Error! Bookmark not defined.***

Table of Figures

Figure 1: ROI Analysis Graph *5*

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 3.0 of the FED document for DCP submission. Major changes since last version:

- Mistakes from the previous version was fixed.
- Update ROI chart and provide a reasonable data to support it.

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

2.1.1.1 Maintenance costs

Our system will have two categories of maintenance. The first category is the software maintenance which includes updates and changes to the source code, and the second category involves maintenance of the forum (i.e. moderation of posts) and keeping the website updated. For simplicity we will refer to the latter maintenance category as *moderation* in this section. For the first category we estimate 100 hours per year with a 10% increase per year. For the second category we have estimated the required time based on a number of assumptions:

1. The effort spent on moderation is directly correlated to the number of active users in the system, as the number of posts requiring moderation on the forum should also correlate to the number of active users.

2. Every user will create on average 50 posts on the forum per year.
3. The average portion of posts requiring moderation will be 10%.
4. The average time spent moderating one post is 5 minutes.
5. The average time for keeping the website updated is estimated to be 52 hours per year with a 10% increase each year.

These estimates are based on consensus among team members and are considered to be conservative expert judgment.

Table 2: Maintenance Cost per Year (hours)

Year	Number of active users	Forum moderation	Website maintenance	Source code maintenance	Total cost
1	2000	833.33	52.00	104.00	989.33
2	4000	1666.67	57.20	114.40	1838.27
3	6000	2500.00	62.92	125.84	2688.76
4	8000	3333.33	69.21	138.42	3540.97
5	10000	4166.67	76.13	152.27	4395.07

2.1.2 Hardware and Software Costs

There are no hardware and software costs because we will select only free software and webhosting.

2.2 Benefit Analysis

The benefit of the system is measured by the number of hours the USC staff has to spend answering questions from students (support-hours). In the below calculation, we use the number of students we expect will join our system, and predict how many support-hours these students would typically account for on a yearly basis. We calculate the required amount of support-hours for the situation where our system is not in place and for the situation where the users are using our system. The difference between these numbers will be the total hours saved by USC staff per year. Our estimates are based on the following assumptions:

1. 10% of students will make on average one support request per week.
2. Each support request will have an average resolution time of 20 minutes.
3. If a student uses our system, his probability of making a support request is reduced by 50%.

Note that in our model, the number of users increase each year. This number is likely to flatten out as the number of users reaches the total number of enrolled students.

Table 3: Benefits per Year (hours)

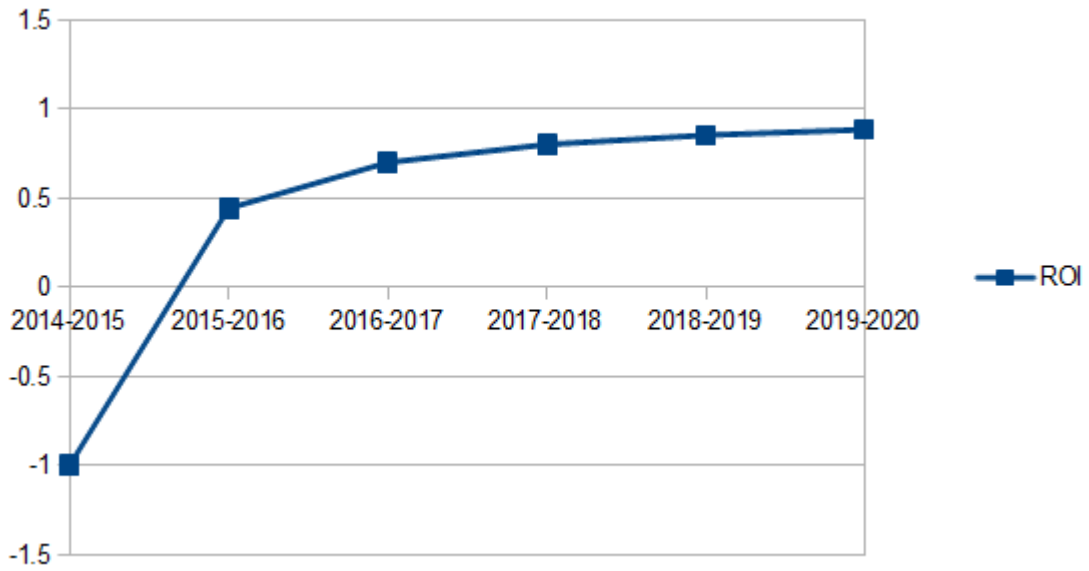
Year	Number of active users	Support hours per year if no system exists	Support hours per year with system (50% reduction)	Hours saved per year for support
1	2000	3432	1716	1716
2	4000	6864	3432	3432
3	6000	10296	5148	5148
4	8000	13728	6864	6864
5	10000	17160	8580	8580

2.3 ROI Analysis

Table 4: ROI Analysis

Year	Cost	Benefit (Effort Saved)	Cumulative Cost	Cumulative Benefit	ROI
2014-2015	200	0	200	0	-1
2015-2016	989.33	1716	1189.33	1716	0.44
2016-2017	1838.27	3432	3027.60	5148	0.70
2017-2018	2688.76	5148	5716.36	10296	0.80
2018-2019	3540.97	6864	9257.33	17160	0.85
2019-2020	4395.07	8580	13652.40	25740	0.89

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 shows that more than 80% of users 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 has 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	Software/Technology used: Laravel, MySQL

System	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 capabilities 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.

4. Process Feasibility

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

Table 7: 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. Because we will build 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 require no cost of ownership because we will use only open source software and free service.
Not-so-powerful local	High	High	We have minimal cost and

machines			we also have no infrastructure right now. We will be using free left over 8 year old laptops.

5. Risk Assessment

Table 8: Risk Assessment

Risks	Risk Exposure			Risk Mitigations
	Probability Loss	Potential Magnitude	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. Advertise 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	8	6	48	Prioritize the prototyping. Use minimum effort for each prototype to gain enough information to see the feature is feasible.

almost all of them require 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 Requirements 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

In our project, we utilize evaluation matrix based on the listed criteria and we finalized NDI/NCS items in our system. We selected PHP as our web programming language because it is familiar for most of our teammates. Laravel is our PHP framework. We find out Laravel has strong community and relatively robust documentation, Moreover, its development time is short in comparison to J2EE framework. Apache will be our web server. MySQL will be our DMBS. For the front-end, we will use Bootstrap. For interoperability, our NDI/NCS items operate well and do not have conflicts with each other.

6.1.1 COTS / GOTS / ROTS / Open Source / NCS

Table 9: NDI Products Listing

NDI/NCS Products	Purposes
Laravel	PHP Framework, It provides 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

In WAT(We Are Trojans) network 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

There is no legacy system.

6.2 Evaluation Summary

Table 10: NDI Evaluation

NDI	Usages	Comments
Apache (2.4)	Web Server	**Positive points** - Freeware - Widely used - Documentations available **Negative points** - No negative points
MySQL(5.0)	Database	**Positive points** - Freeware - Widely used - Documentations available - Suitable for Large scale system - Good performance **Negative points** - No negative points
Laravel	PHP framework	**Positive points** - Freeware - Widely used - Robust documentations - Easy to learn and understand framework - Robust community - Clean Framework **Negative points** - No negative points

Bootstrap	CSS framework	**Positive points** - Freeware - Widely used - Robust documentations - Nice UI components **Negative points** - No negative points
JQuery	Javascript library	**Positive points** - Freeware - Widely used - Powerful components **Negative points** - No negative points