

REFLECT Mobile

EOSP, Summer 2014

Team Rewyndr:
Zakhar Herych
Pranav Saxena
Ulyana Skladchikova
Mengyan Wang

Team



Zakhar
Herych



Mengyan Wang



Ulyana
Skladchikova



Pranav
Saxena

Mentor

Matthew Bass,
Associate Director of Software
Engineering Professional
Programs for Corporate and
Alumni Relations



Stakeholders



David Palmer,
founder of Rewyndr

Goal: create a social network for small communities and allow them share their experience through rich interaction with photos

REFLECT
REFLECT

REMEMBER THIS MOMENT?

Cleary 7th Floor 2001 > Graduation Day



EXPLORE THIS MOMENT

WHEN WAS THIS? MAY 8, 2005

WHO'S HERE? CHRIS BAKER, PAIGE LARSON, KEVIN ROGERS, CHRISTINE WANG

WHAT DID THEY SAY?

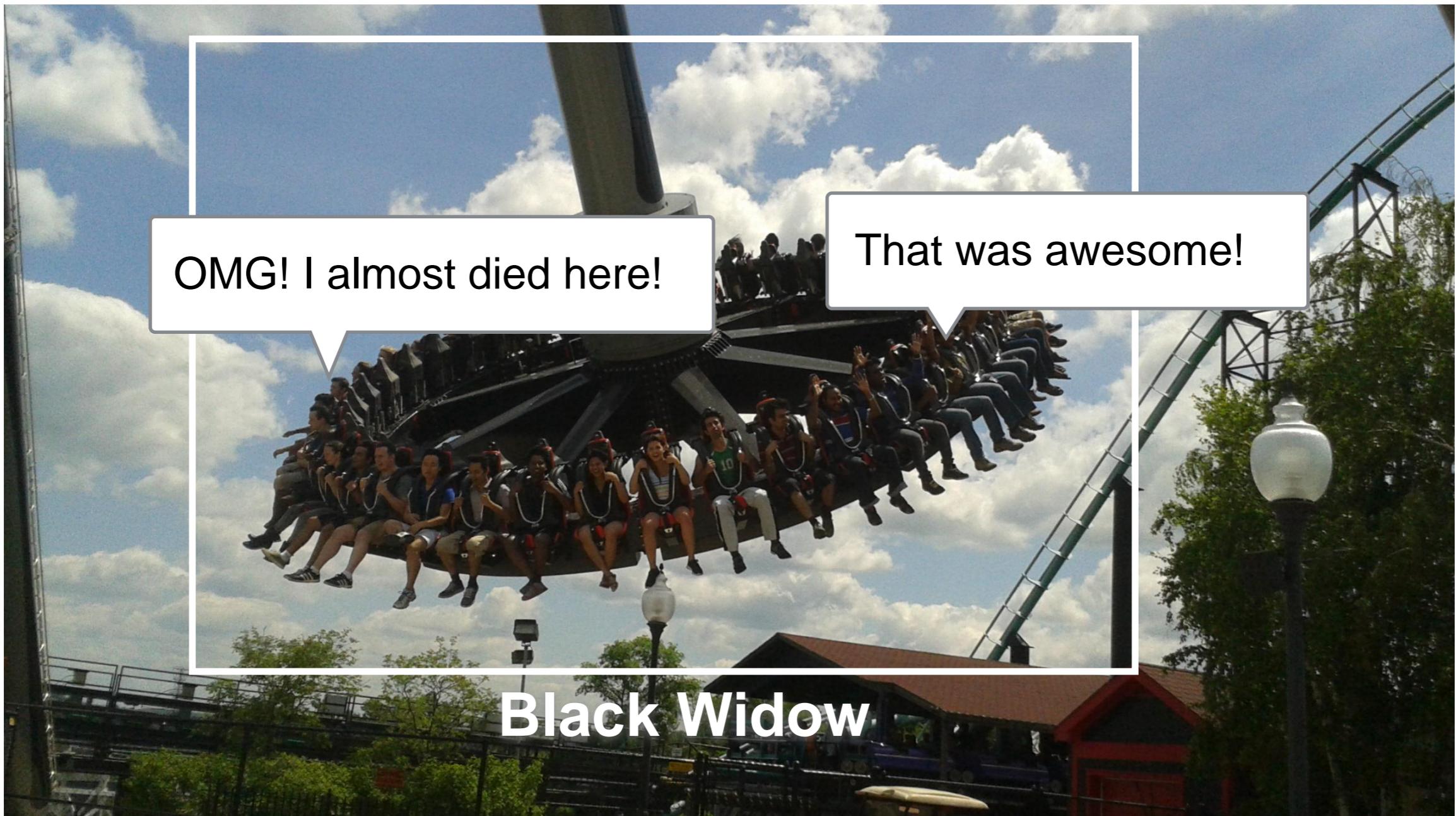
★ CHRISTINE WONG WAS FEELING short!!
MATT WINTHROP

Context



Community

What problem do we solve?



Black Widow

Moment

Web prototype

The image shows a web prototype for a social platform. At the top, there is a green navigation bar with the word "REFLECT" in white on the left, and "My Profile", "My Communities", and "Logout" in white on the right. Below the navigation bar, the main content area has a light gray background. On the left, the text "My Communities" is displayed in a large, dark green font. On the right, there is a "CREATE NEW COMMUNITY" button in a dark green box. The main content area contains three cards, each with a small image and a title: "MSIT ALUMNI 2014" (image of a woman holding a pizza), "MY FAMILY" (image of a smiling man), and "CAT LOVERS" (image of a fluffy cat). Each card has a small circular icon in the top right corner.

REFLECT

My Profile My Communities Logout

My Communities

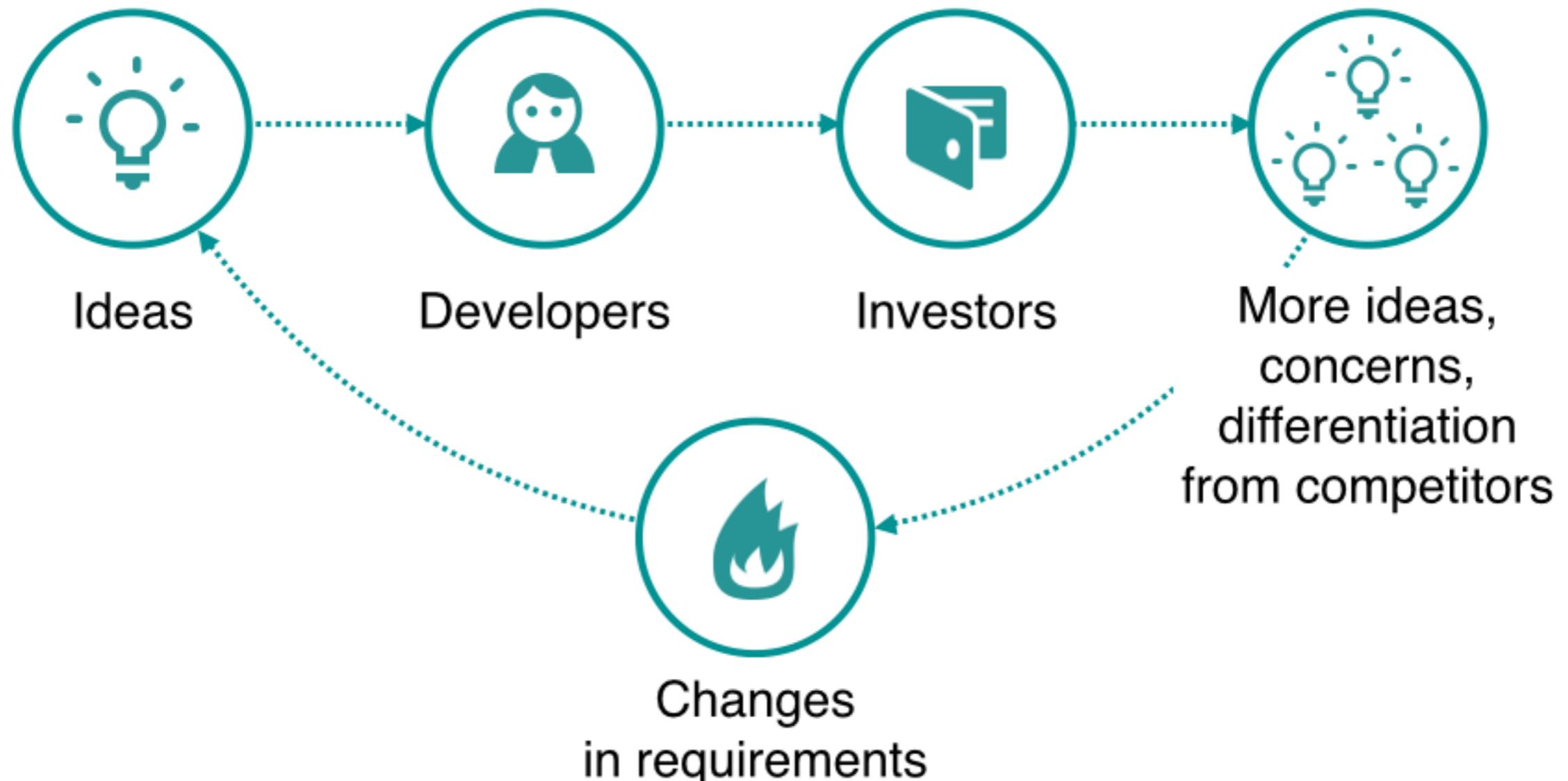
CREATE NEW COMMUNITY

MSIT ALUMNI 2014

MY FAMILY

CAT LOVERS

Business Context



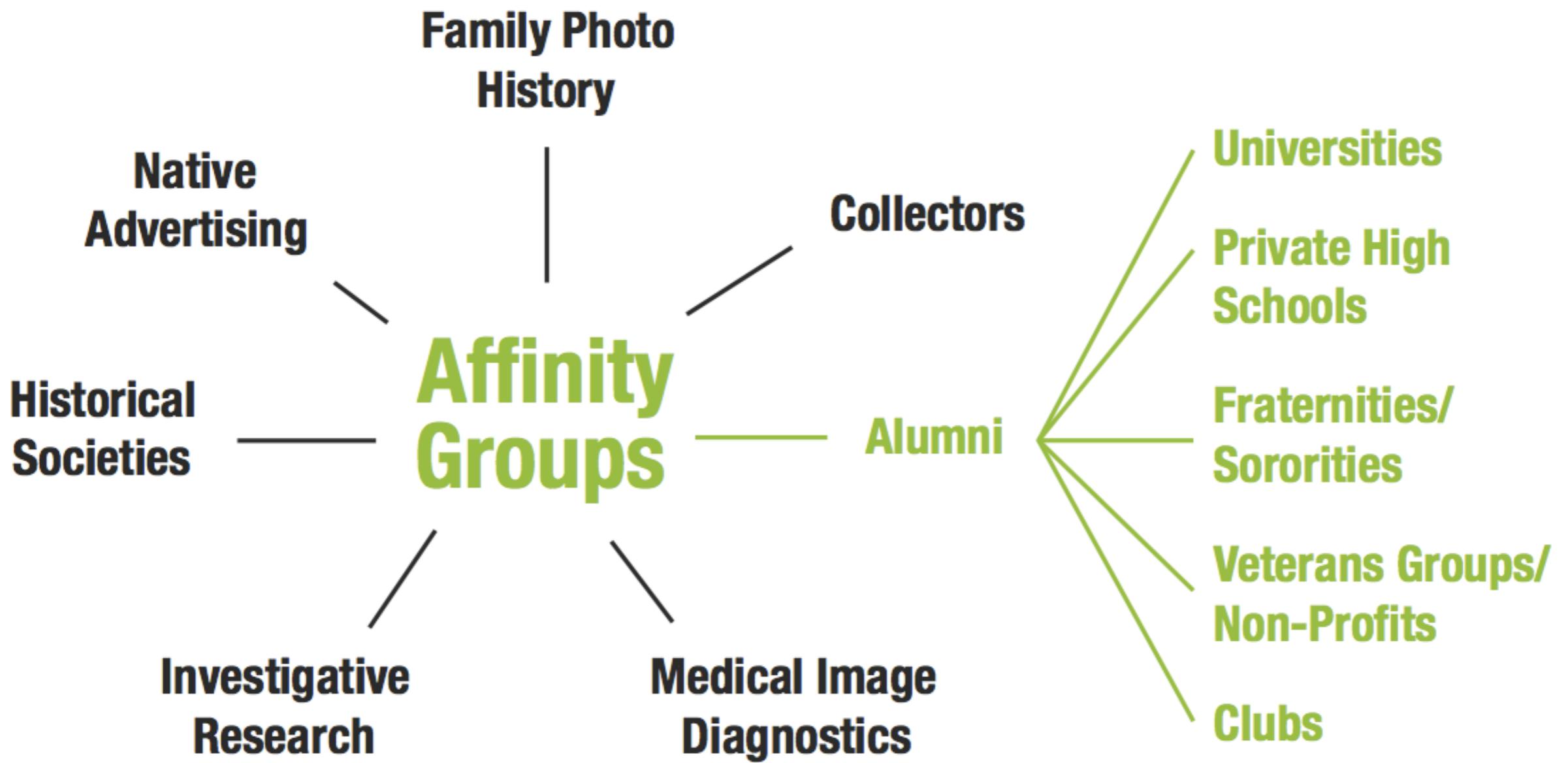
⌚ Funding comes in small increments

Project Phases

- Initial phase – define the product-to-be
- Design phase
- Construction phase

Initial Phase

Market opportunities



Initial Phase

No requirements set upfront.

Possible directions for MSIT-SE project:

- Capture new platforms (mobile)
- Capture new markets:
 - collectors
 - museums
 - historical archives
- Improve existing web product (utilize object recognition for tagging and search)

Initial Phase

Evaluate requirements from different perspectives:

- business value
- cost of development
- technical feasibility

Trade-off analysis

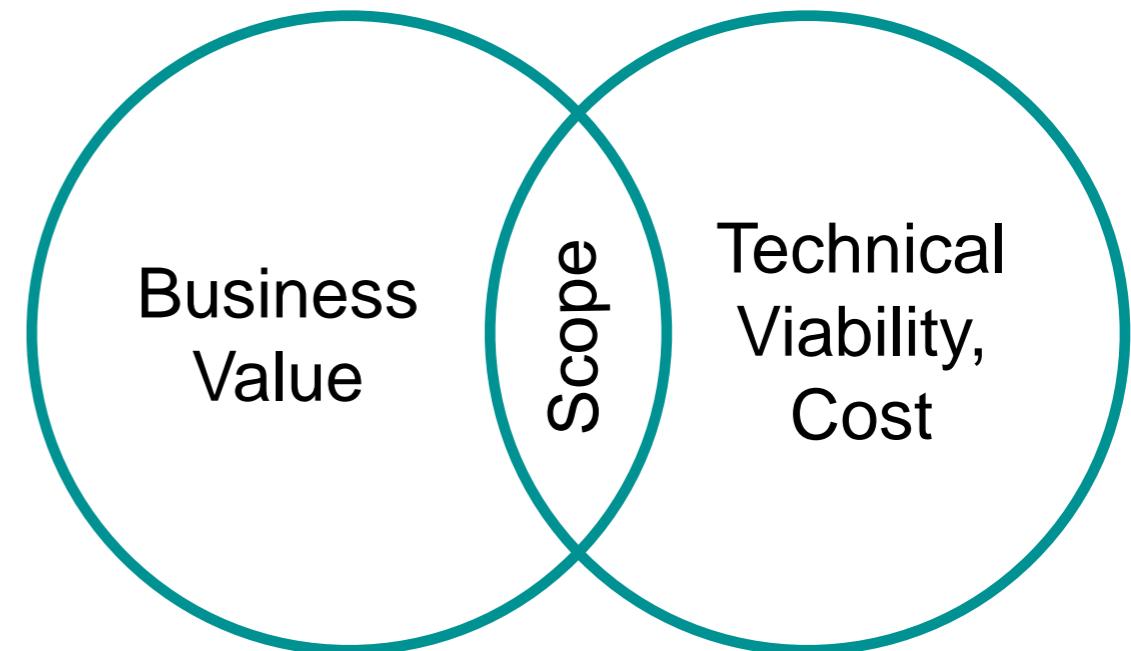
Technical analysis:

Analysis of maturity of algorithms

Business analysis:

User-centered design: personas, goals

Cost and business value analysis



Results: What can be done without over-commitment?

Mobile application that ports all functionality of the existing web prototype to mobile devices and utilizes computer vision techniques for automated tagging.

Initial Phase

Distilled goal for MSIT-SE project:

Create functional prototype of the mobile application to showcase to venture capitalists to gain funding for the next phases of product development.

Initial Phase – Reflections

We had to:

- Go through Initial phase
- Introduce traceability between decisions:
“won't haves”
- Establish set of time-boxed practices to quickly investigate each possible direction from many perspectives
- Drop practices that were not appropriate

Design Phase - Risks

Initial Sources of risks:

Technology, Changing Requirements, Market, Resources

Example:

Source of Risk (category):

Technology

1. Condition:

Immature computer vision algorithms

Consequence:

- Algorithms do not meet satisfactory quality attributes
- Object boundary detection can't be implemented in given time

Mitigation strategy:

Early prototyping, time-boxing, benchmarking.

Experiments phase within ACDM to estimate the quality.

Impact: High, Probability: High

2. Condition:

- Existing backend tier doesn't support required functionality
- Porting image processing functionality to device

Consequence:

- Performance is not feasible on mobile

Mitigation strategy:

Early prototyping, time-boxing, benchmarking.

Impact: High, Probability: High

Design Phase - Process Mgmt.

Why ACDM?

- helps refine functional requirements, quality attributes and constraints
- helps set and maintain expectations of stakeholders
- helps identify technical risks early
- helps extract what is unknown in the project

Why Tailor ACDM?

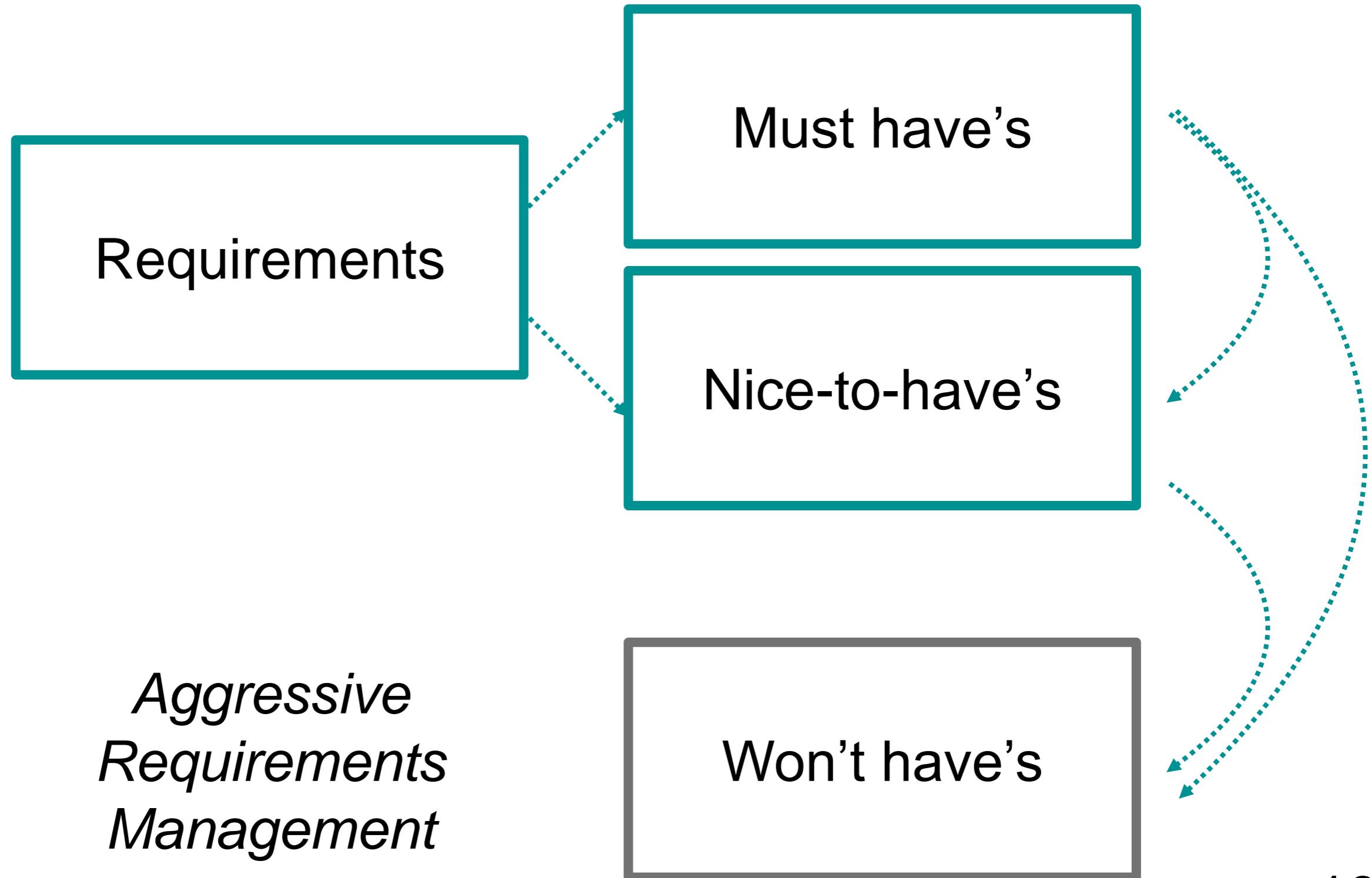
- We had initial requirements upfront from Initial Phase
- Used User-centered design technics
- Production phase of ACDM = Scrum

Design Phase - Planning

- Defined roles
- Planned each design iteration
- Mapped phases to timeline
- Architecture Drivers Workshop
 - Prioritized Quality Attributes
 - Business constraints
 - Technical constraints

Task Name	Duration
▪ Design Phase	55 days
Contextual Inquires, Personas, Scenarios	7 days
Discover Architectural Drivers	6 days
Establish Project Scope	6 days
▪ Design Iteration 1	11 days
Create Notional Architecture	
Review the Architecture	
Do the Go/No-Go Decision	
▪ Design Iteration 2	11 days
Plan Experiments	
Execute Experiments	
Refine the Architecture	
Do the Go/No-Go Decision	
▪ Design Iteration 3	11 days
Plan Experiments	
Execute Experiments	
Refine the Architecture	
Do the Go/No-Go Decision	
Production Planning & Design	7 days

Design Phase - Requirements Mgmt.



Design Phase - Requirements Mgmt.

Must-have requirements	Priority	Nice-to-have requirements	Priority	Out of scope requirements
Android Mobile version of REFLECT	0	Voice-to-text	2	Gaming and Incentive Systems
Audio recording or uploading	2	Face detection	3	Face recognition
Object boundary detection and categorization	1	Event Planning functionality	4	Object Recognition
Alerts on activity relevant to user	3	Search by key words, Boolean Search	1	Search for similar objects based on computer-vision technology, enhanced search and discovery capabilities
On-line donation	4			Tagging frames in videos
				Messaging between users

Design Phase – Architecture

1) Performance (in terms of):

A. responsiveness of the UI

(longer computation task does not block the application and runs in the background)

B. speed of boundary detection algorithm

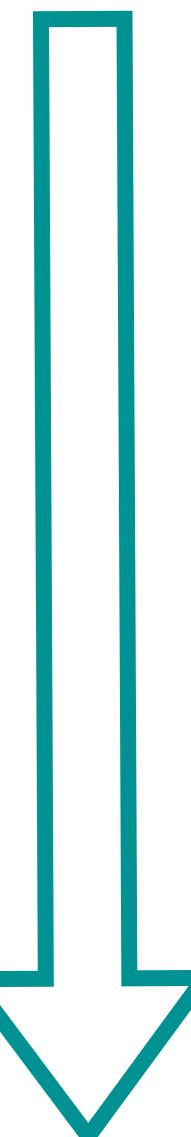
C. response time of uploading a file/image - network

latency, size of the file (inbuilt image compression to reduce the resolution of the image).

2) Interoperability - Consistency in the data between the mobile and web application.

Example - Photo uploaded from the mobile application should be reflected in the web application and vice versa for a given user.

DECREASING
PRIORITY



Design Phase – Architecture

Experiments:

- Async Communication Model
- Boundary Detection Algorithm
- Remote API Testing

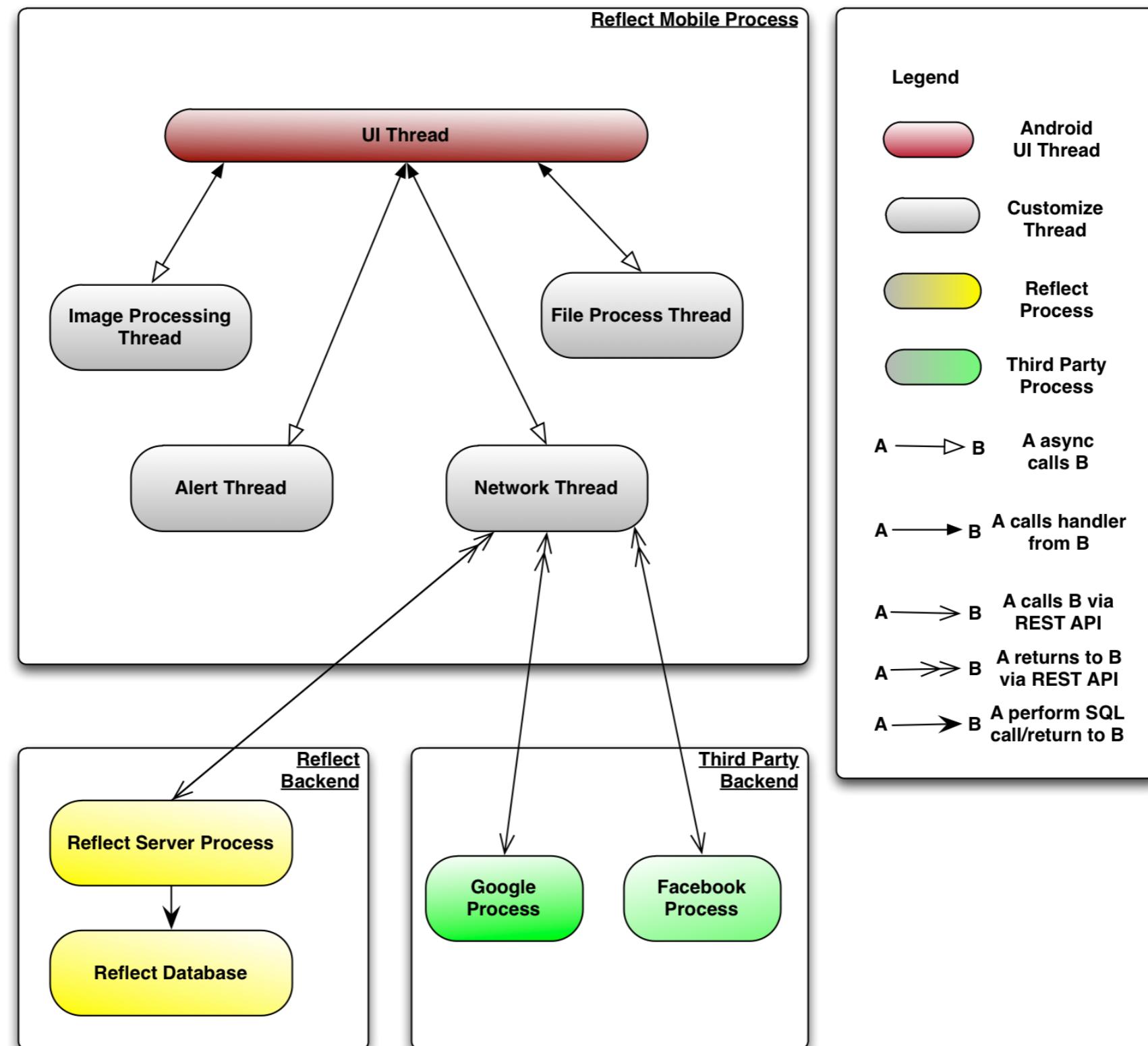
Image Resolution	Time (Second)	Memory (MB)	Image size	Connection type	Time to download
320*240	0.65	<5MB			
640*480	1.52	<5MB			
800*600	2.28	<5MB			
1024*768	3.02		114 KB	3G	0.9s
1280*960	4.21		114 KB	WiFi	0.4s
2048*1536	5.02		75 KB	3G	2s - 4s
			75 KB	WiFi	0.4s

Design Phase – Architecture

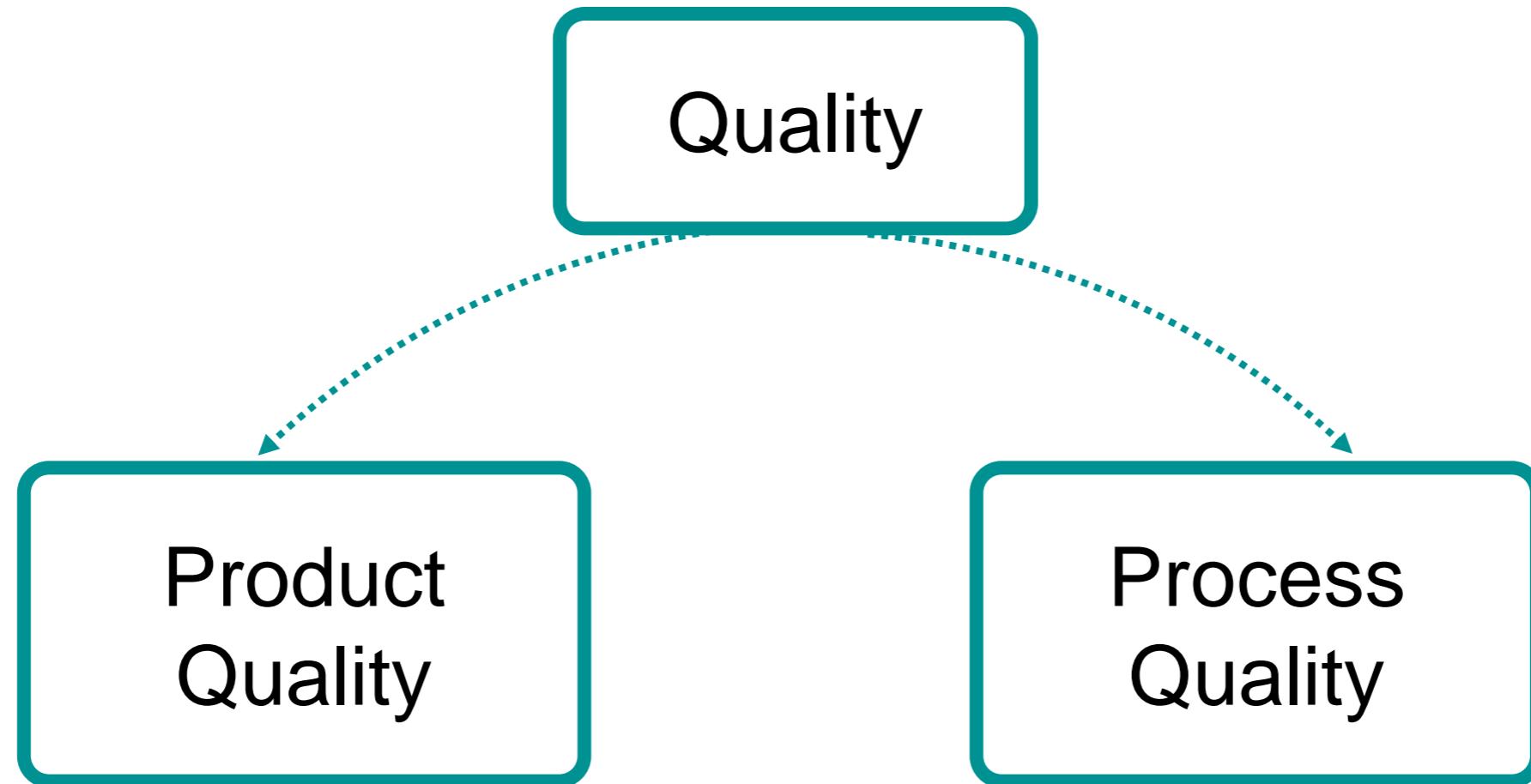
Top Architectural Decisions

- Asynchronous Model of communication
 - Promote Responsiveness - UI thread makes async function call to the Network Manager and Image Processor which in turn invokes async callbacks of the UI thread.
- Allocating responsibility for image processing to the client
 - Technical Constraint - No control of backend
 - Performance on client side can still satisfy QA
 - Reduced Development Effort

Design Phase – Architecture



Design Phase – Quality Management



Product quality goals:
usable prototype

Process quality goals:
Compliance to the
established process

Design Phase – Quality Management

Product Quality

Prototyping and Implementation:

- Review of Experiments results
- User Studies on Prototypes

Artifact Quality

Reviews for documents:

- Scope and Requirements Specification Review
- Architectural Design document Review
- Review of Experiment plans
- Team feedback
- Mentor Review and approval

Architecture Review

Design Phase – Quality Management

Process Quality

- Reflections and team feedback on Process
- Process conformance checklist
- Review with Mentor and Mentor's feedback

Design Phase – Quality Management

	Quality Activity	Project Activity
ACDM Process Compliance	Project Plan Review	Project Planning
	Requirements V&V	Requirement Gathering and Elaboration
	Risk Management	
	Verify Architectural drivers	Discover AD
	Verify Statement of Work	Establish Project Scope
	Analyze Notional Architecture for feasibility	Create Notional Architecture
	Architecture Review	
	Revise Plans	Experiments Planning
	Review of experiments results	Experiments Execution

Design Phase – Configuration Mgmt.

- Artifacts management:
 - Established artifacts internal review process
 - Established customer validation process for requirements artifacts
 - Established formal process for monthly cleaning and reorganizing artifacts
 - Established artifacts repository
 - Clearly defined categories of the artifacts to improve coordination
- Task management:
 - Established templates for weekly reports over tasks that were planned and done
 - Established the process for tasks assignment and status tracking
 - Installed and compared tools that help track the team activities
- Code management:
 - Created repository for storing the code from experiments and early prototype

Design Phase – Tracking

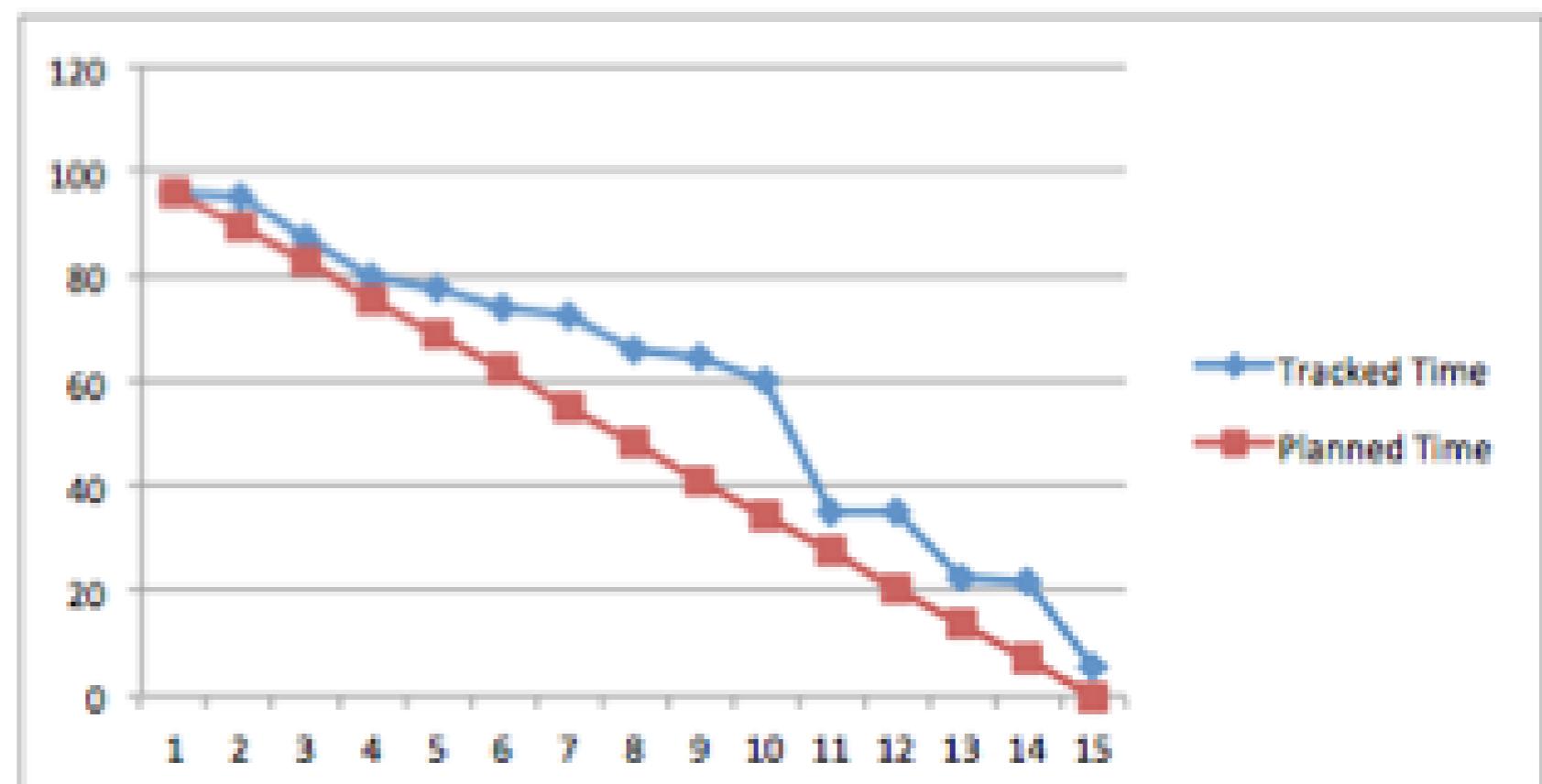
Results:

24 User Stories

6 Refined Quality Attributes

3 Iterations of architectural detailed design

- ✓ Time planned
vs
Time spent
- ✓ Finished tasks



Iteration #3

Design Phase – Risk Mgmt.

Risk Management Process

1. Assigned role of Risk Manager
2. Initial risk elicitation meeting that included
 1. Preparation
 2. Brainstorming on ideas
 3. Categorization of risks
 4. Risk likelihood level and impact
 5. Mitigation strategies
 6. Contingency plan
 7. Convert mitigations into tasks and assign them
3. Bi-weekly (each Design iteration) risks tracking meeting.
4. Monitor risks to understand if they became problems

Design Phase – Reflections

Project Planning

Simply defining tasks and assigning responsibility is not enough for complex tasks

Wrong: Executing complex tasks without plan is hard and you don't necessarily know how to do the task.

Correct: For the experimentation phase we followed a process for creating experiments plans upfront. These plans helped us to better understand the tasks that team members were not familiar with and had a lot of unknowns in nature.

Design Phase – Reflections

Project Tracking

Just tracking time is not enough.

Wrong: The only information that we get from such tracking is that tracked time doesn't match planned time.

Correct: Instead we should have tracked the effort spent on certain tasks and activities. That would have helped us to:

- Track team velocity
- Understand whether team is on track
- Do better estimation during planning

Design Phase – Reflections

Context & Requirements Management

No meetings without agenda

Wrong: the unstructured process for customer communication allowed us to receive a lot of information that was irrelevant to our final goals.

Correct: following a structured process for customer interaction on a weekly basis would help us make the meetings more productive and elicit requirements from customer communication earlier.

Good news:

Introduced variations in requirements and their relative priorities to the point where they are not significant to architecture -> allowed to keep agility + mitigate technical risks

Design Phase – Reflections

Architecture

For the projects with high level of technical risk using ACDM-like process is very beneficial

Wrong: without understanding the unknowns, we wouldn't be able to focus on the right systemic properties of the project.

Correct: adopting a process that uncovers the quality attributes and allows building an architecture with required systemic properties significantly reduces the level of technical risk in the project.

Design Phase – Reflections

Risk Management

Risk mitigation activities should be practiced continuously

Wrong: Our team identified risk of team members being unavailable at certain time frames. Our mitigation strategy was to conduct weekly work sessions.

However, we practiced this only for several weeks and therefore were behind the schedule by the end of semester.

Correct: We should have continued the execution of mitigation strategy until the risk is mitigated. That would allow us to prevent this risk from becoming a problem.

Construction Phase – Process Mgmt.

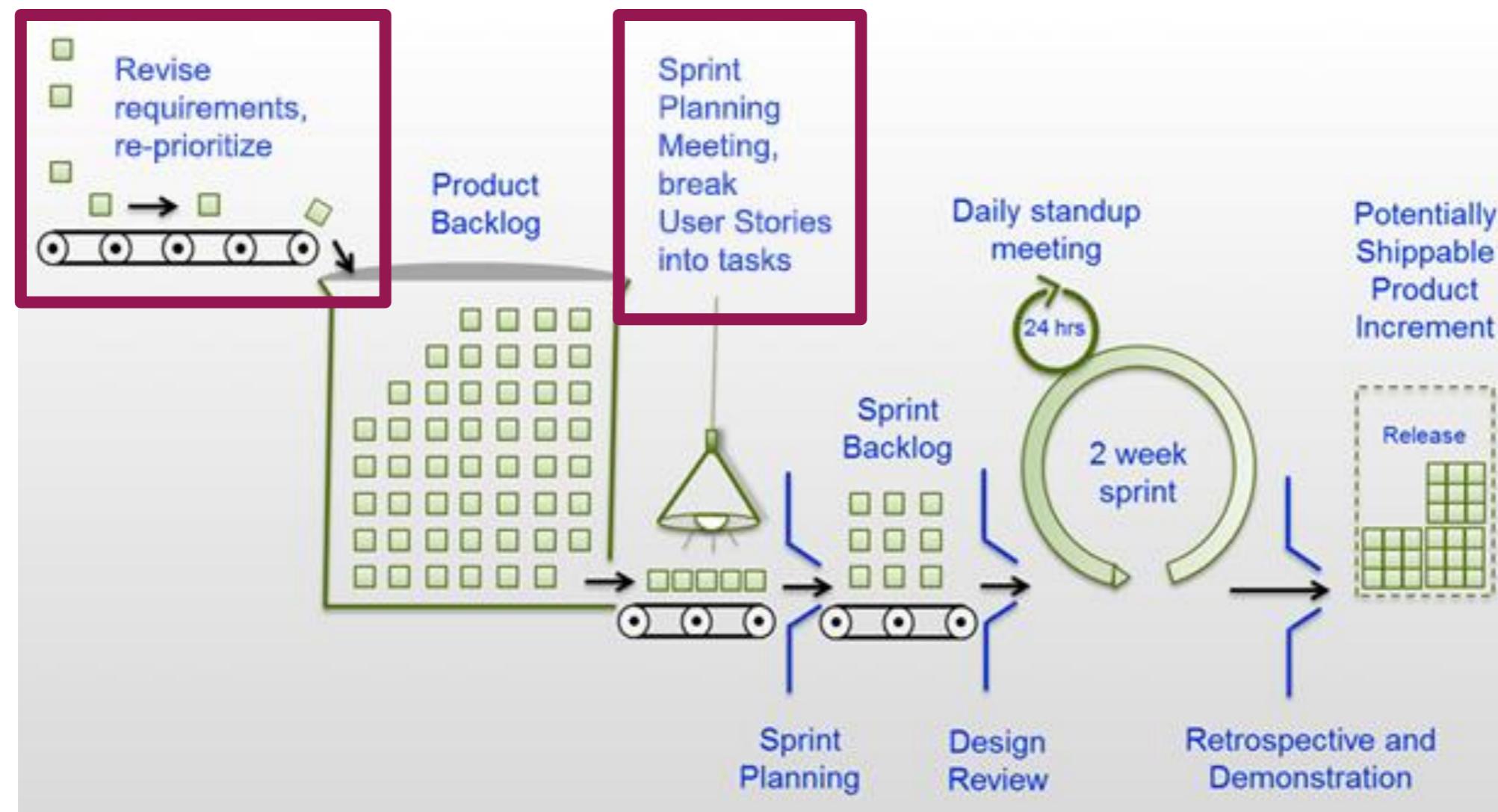
Why Scrum?

For development phase Scrum seemed suitable for our needs:

- Small team
- Adaptable to both technical and business changes
- Constant communication with stakeholders

Construction Phase – Process Mgmt.

Adoption of Scrum:



Construction Phase – Planning

6 Sprints – First sprint for training, the rest of the sprints for developing and testing

4 Releases – Finish partial user stories in each release and demonstrate to the customer

Sprint	Date	Release
Sprint 1	May, 19 - June 1	Trial sprint, no release
Sprint 2	June, 2 - June, 15	Release 1
Sprint 3	June, 16 - June, 29	Release 2
Sprint 4	June, 30 - July, 13	Release 3
Sprint 5	July, 14 - July, 27	Release 4
Sprint 6	July, 28 - August, 10	Hotfix

Construction Phase – Sprint Planning

User Stories

Order by

REW-129

[View Photo](#)

REW-92

[Edit detail](#)

REW-91

[Edit story](#)

REW-90

[Edit tag](#)

REW-89

[Add sound](#)

REW-88

[Add detail](#)

REW-87

[Add story](#)

REW-86

[Add tag](#)

REW-85

[List details, stories and sounds related to photo](#)

Tasks

Description

As a user I want to attach detail to a region of a photo or the whole photo

Sub-Tasks

1. add detail: design

2. Add detail: activity and view

3. add detail: network

4. add detail: code review

Activity

All

Comments

Work Log

History

Activity

There are no comments yet on this issue.

Time

Watchers:

1 Start watching this issue

Dates

Created:

21/May/14 6:43 PM

Updated:

30/Jun/14 11:22 AM

Resolved:

26/Jun/14 7:15 PM

Time Tracking

Estimated:

6h

Remaining:

0h

Logged:

6h 34.5m

Include sub-tasks

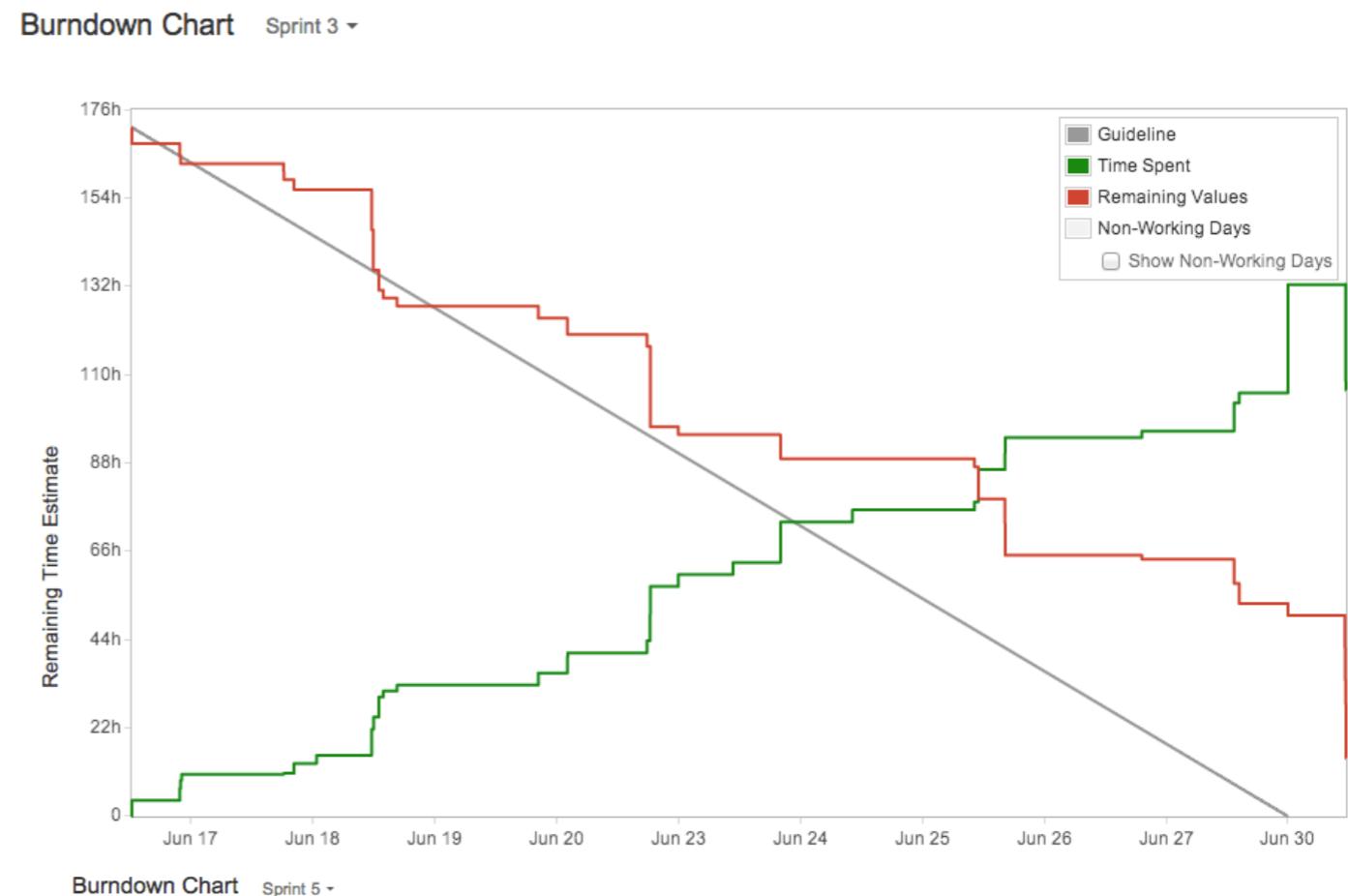
Construction Phase – Project Tracking

Time +User Stories

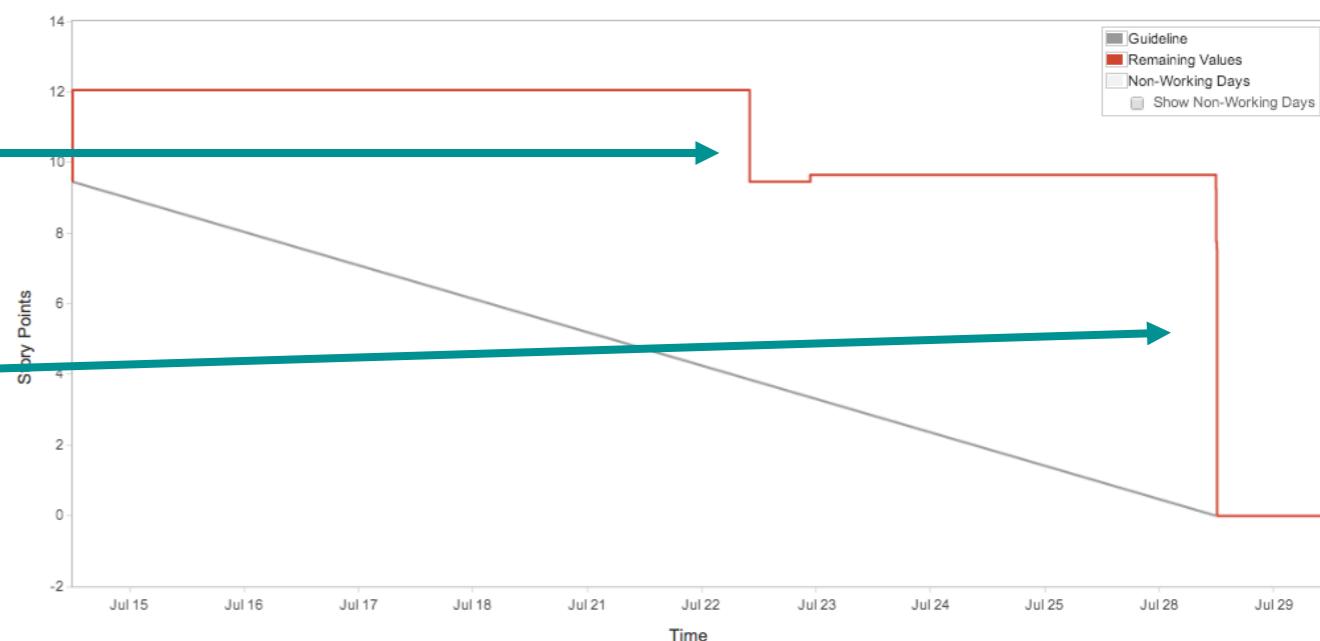
Burndown Charts
Cumulative Diagram

Mid-Sprint review

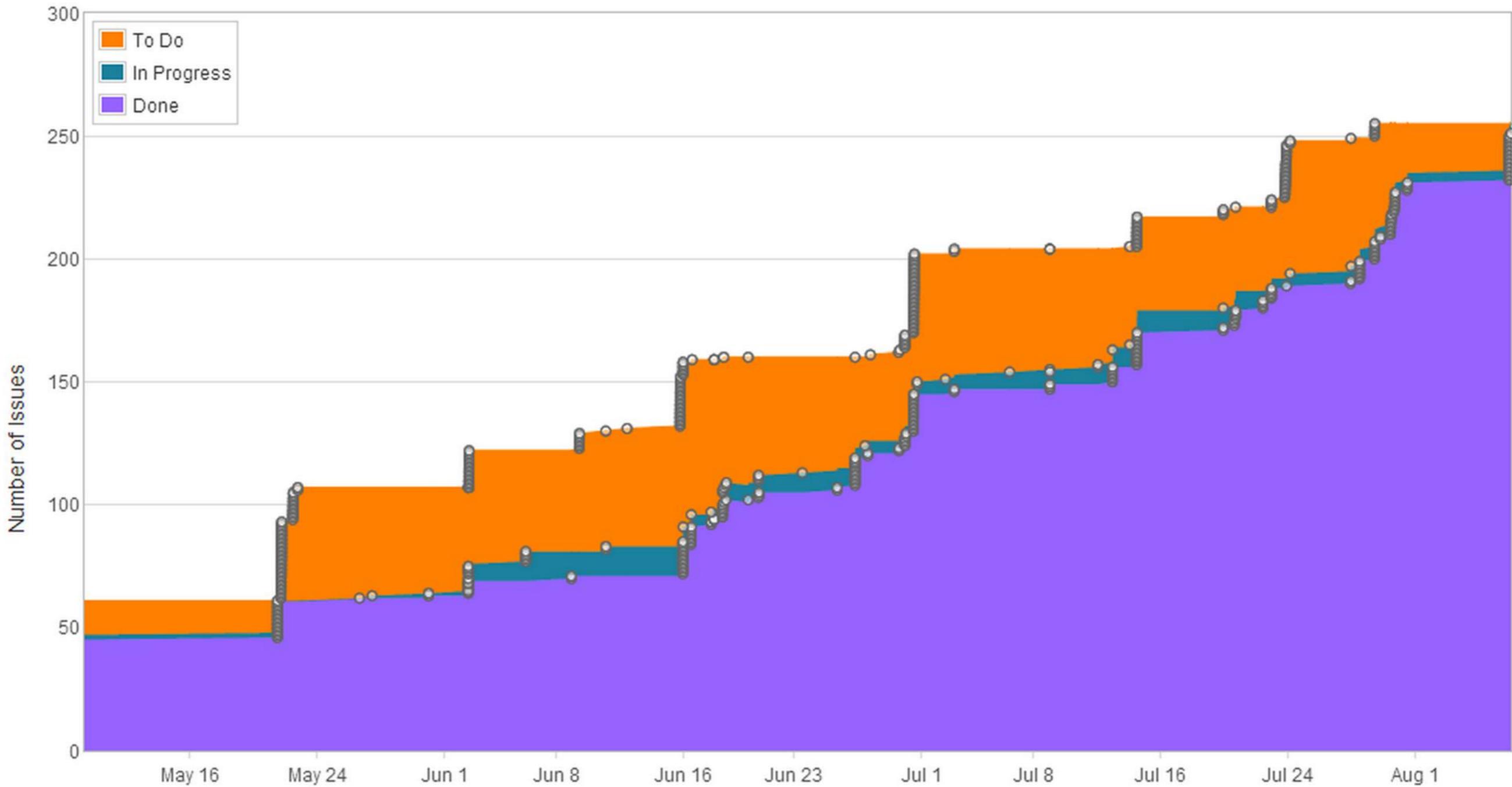
Sprint review



Burndown Chart Sprint 5 ▾



Construction Phase – Project Tracking



Construction Phase – Requirements Mgmt.

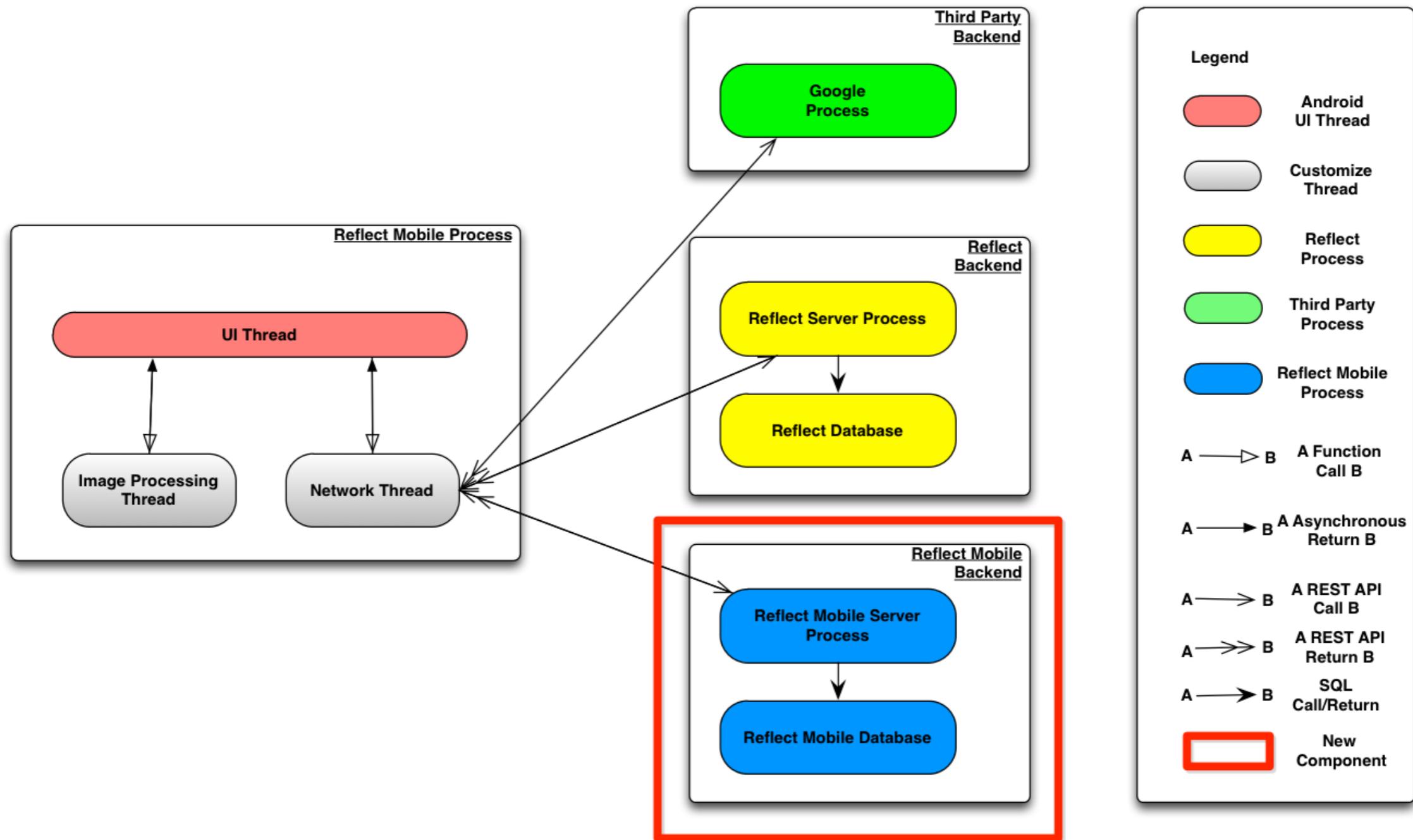
- Continuous Customer Communication
- Verify requirements when divide User Stories into tasks

Technical risks became problems →

! Develop lightweight backend to store tag and sound metadata

- Result: Architecture evolved

Construction Phase – Upd. Architecture



Construction Phase – Upd. Architecture

- Trade-offs

Second backend → Decreased maintainability

- Quality attributes

Performance → The second backend should return the metadata of a picture in 100 ms.

Design Phase – Architecture

Performance

Image Resolution	Threshold (Pixel)	Memory (MB)	Time (Second)	Required (Second)
320*240	30	<5MB	0.35	0.4
640*480	30	<5MB	0.55	1
800*600	30	<5MB	0.95	1.6

Image size (KB)	Connection type	Time to download (Second)	Required (Second)	Time to upload (Second)	Required (Second)
215 KB	3G	0.47	0.5	0.56	0.7
215 KB	WiFi	0.25	0.3	0.38	0.5
89 KB	3G	0.21	0.3	0.29	0.4
89 KB	WiFi	0.13	0.2	0.18	0.2

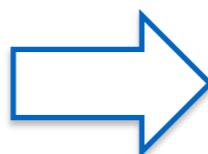
Construction Phase – Conf. Management

Initial setup of the development configuration

Environment setup:

- IDE + Emulator
- Github
- JIRA
- Google doc

Commit and merge
at the end of sprint



Strict, structured policy
for commit and merge

Construction Phase: Configuration Management

Commit policy.

The procedure goes as follows:

- 1) Team member is assigned a feature
- 2) Team member creates the code for the feature
- 3) Team member tests the solution and fixes the discovered issues
- 4) Team member checks the code for compliance to coding standards
- 5) Team member commits the new changes with descriptive message (definition of done)
- 6) Team member pulls the code from the master branch
- 7) Team member resolves all the merge conflicts if any
- 8) Team member pushes his changes to the master branch
- 9) Assign code reviews to other teammates

Construction Phase – Quality Mgmt.– Testing Strategy

Functional Testing

User Interaction Testing

Session-based Exploratory Testing

Script-based Manual Testing

Non-functional testing

Usability Testing

Performance Testing

Test responsiveness of the UI

Test speed of boundary detection algorithm

Test response time of uploading/downloading a file/image

Unit Testing

Testing exceptions

Testing memory leaks

System Testing

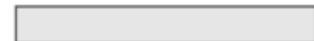
Integration Testing

Installation/Deinstallation Testing

Construction Phase – Quality Mgmt

Clover Coverage Report ReflectMobile Coverage Report Coverage timestamp: Wed Aug 6 2014 23:21:02 EDT	project stats:	LOC: 10,155	Methods: 520
	NCLOC: 7,448	Classes: 83	
	Files: 35	Pkgs: 7	

	Branch	Stmt	Method	Total
Clover database Wed Aug 6 2014 23:12:10 EDT	76.3%	85.4%	83.8%	84% 

Packages	Branch	Stmt	Method	Total
com.google.android.gms	-	-	-	- 
com.reflectmobile	-	-	-	- 
com.reflectmobile.activity	75.9%	83.3%	79.2%	81.8% 
com.reflectmobile.utility	53.8%	85.5%	100%	83.2% 
com.reflectmobile.data	84.2%	86.1%	88.7%	86.5% 
com.reflectmobile.view	75%	93%	92.9%	91.1% 
com.reflectmobile.widget	92.2%	98%	90.6%	96.5% 

Construction Phase – Quality Mgmt

Testing UI Checklist

Verify all alignments and sizes of all elements. Alignment and sizes should be proper/same as defined in wireframe.

Spell check all text in all screens. There should not be any errors.

Verify that Font sizes are consistent. Font sizes for headers, input fields, all labels should be consistent, refer to wireframes. Ensure that the screen font family matches from screen to screen.

Verify that any kind of text is cut off.

Verify that any kind of graphics is not blurred, distorted.

Verify that there is visual feedback when response to any action takes more than 3 seconds.
If the screen takes more than 5 seconds to display the results/page, it should contain a progress bar so that the user understands the processing is continuing.

Verify that each screen should be visible for the time necessary to comfortably read all its information by moving between screens of an application. Each screen must be visible for the time necessary to comfortably read all its information.

Verify that error messages in the Application are clearly understandable. Ensure that error messages are informative, grammatically correct, and not condescending.

Verify that error messages clearly explain to a user the nature of the problem, and indicate what action needs to be taken (where appropriate).

Verify that any function selected in the Application should start within 3 seconds.

Verify that If the screen contains text boxes that allow data entry, ensure that the width of data entered does not exceed the width of the field.

Construction Phase – Quality Mgmt.

Details

Type: Bug Status: **RESOLVED** (View Workflow)

Priority: Trivial

Labels: None Documentation Coding Standard

Sprint:

Defect Type: Activity View (UI) Network Data

Phase Found:

Sprint when defect will be removed:

Defect Types

People

Assignee:  Zakhar Herych [Administrator]

Assign to me

Reporter:  Ulyana Skl

Votes: 0

Watchers: 1 Stop watching this issue

Description

Consider the following flow:

- 1) Tags mode
- 2) Add new Tag
- 3) Put tag
- 4) AFTER THAT EDIT NAME - keyboard is up
- 5) Hit DONE
- 6) "View Tags" screen + keyboard. Keyboard should be hidden.

Attachments



42 bugs found and fixed

Dates

Created: 23/Jul/14 8:34 PM

Updated: Just now

Resolved: 5 days ago

Agile

Active Sprint: Sprint 6 ends 07/Aug/14
[View on Board](#)

Construction Phase – Risk Management

Monitoring Risks

Dealing with Risks that became problems

Source of Risk	Condition	Consequence	Mitigation	Probability	Impact	Status
Development Process (Team Time Resources)	Team members tend to take lot of electives	Team members might be overloaded with electives. This can affect deadlines	Proper planning, buffer period, distribute project resources accordingly	High	High	Problem (Open)
	Team members (resources) tend to search for jobs in this period.	Resource can become unavailable (Interviews / trips / any other errands). This can affect deadlines	Planning to compensate hours devoted to development in case of interviews. Policy for time compensation.	High	High	Problem (Open)
Design	Android components difficult/ impossible to implement	Project might lose value, client may not be satisfied	Scope Exercises, early experimentation	Medium	High	Mitigated
	Due to limited time, team members tend to spend most	This might lead to a situation where the features are not working as expected	Thorough quality plan, rigorous testing required	Medium	High	

Design Phase – Reflections

Configuration Management

Clear Code commit policy should be defined early in the process

Wrong: Not adhering to a structured process (e.g. no strict policies for committing code) led to both time and resource wastage, merge conflicts used to happen.

Correct: Following a well defined process to commit code for respective features helped team members collaborate easily on both dependent and independent features.

Construction Phase – Reflections

Risk Management

Keep tracking of the risks with high impact even it has low possibility.

Wrong: Ignore the risks with high impact but low priority

Correct: Keep tracking this kind of risk

Construction Phase – Reflections

Quality Management

Simply get code coverage is not enough

Wrong: Simply use test tools to get code coverage

Correct: For UI intensive application, manually testing is necessary

**Thank you!
Questions?**

Backup slides

1. Project Planning
2. Project Tracking
3. Process Management
4. Quality Management
5. Configuration Management
6. Architectural Design
7. Risk Management

Planning: Milestone Plan

	Team Reflect Mobile	Client	Subcontractors	Date
Requirements analysis finished and options proposed	O			2/17/2014
Option decision taken		O	O	2/19/2014
Requirements set		O	O	2/26/2014
Requirements created	O			3/31/2014
Architecture drivers elicitation workshop	O	O	O	3/5/2014
Architecture created	O	O	O	4/7/2014
Experiments carried out and documented	O			4/27/2014
Requirements specification elaborated with customer		O		4/2/2014
Requirements specification refined	O			4/6/2014
Requirements sign-off		O		4/9/2014
Architecture refined	O			4/31/2014
Design completed		O		5/4/2014
GUI design first draft	O			4/13/2014
GUI design elaborated with customer		O		4/25/2014

Planning: Milestone Plan



Planning: Experiments

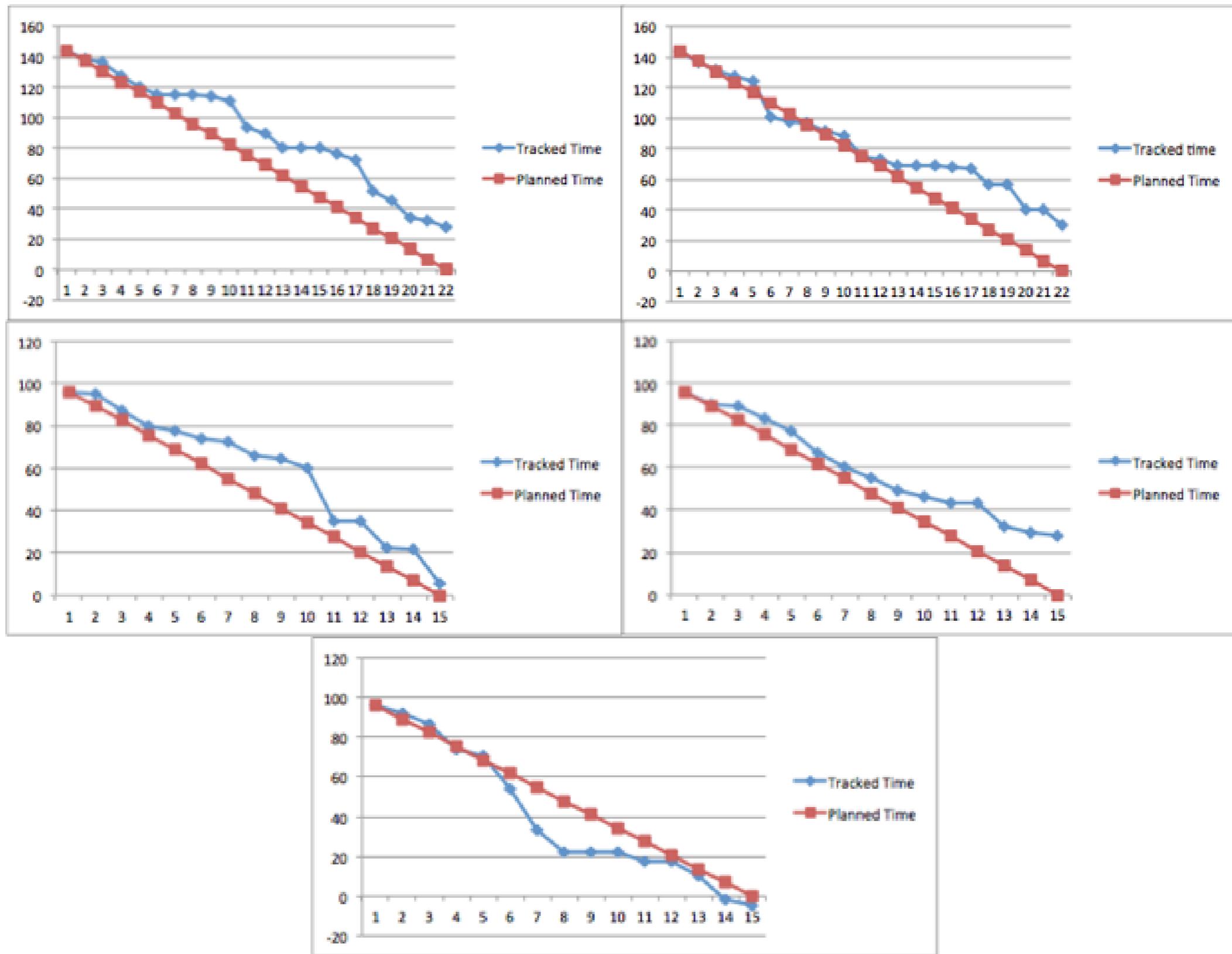
Pre experiment	
Experiment title:	Experiment ID:
Issue ID/description:	Responsible engineer:
Issue deposition:	
Purpose:	
Description of the experiment:	
Artifacts created:	
Completion criteria:	
Resources required:	
Estimated duration and key milestones:	
Post experiment	
Summary of the findings:	
Actual duration:	
Actual resources:	
Responsible engineer's recommendations:	

Design Phase: Tracking

Weekly report

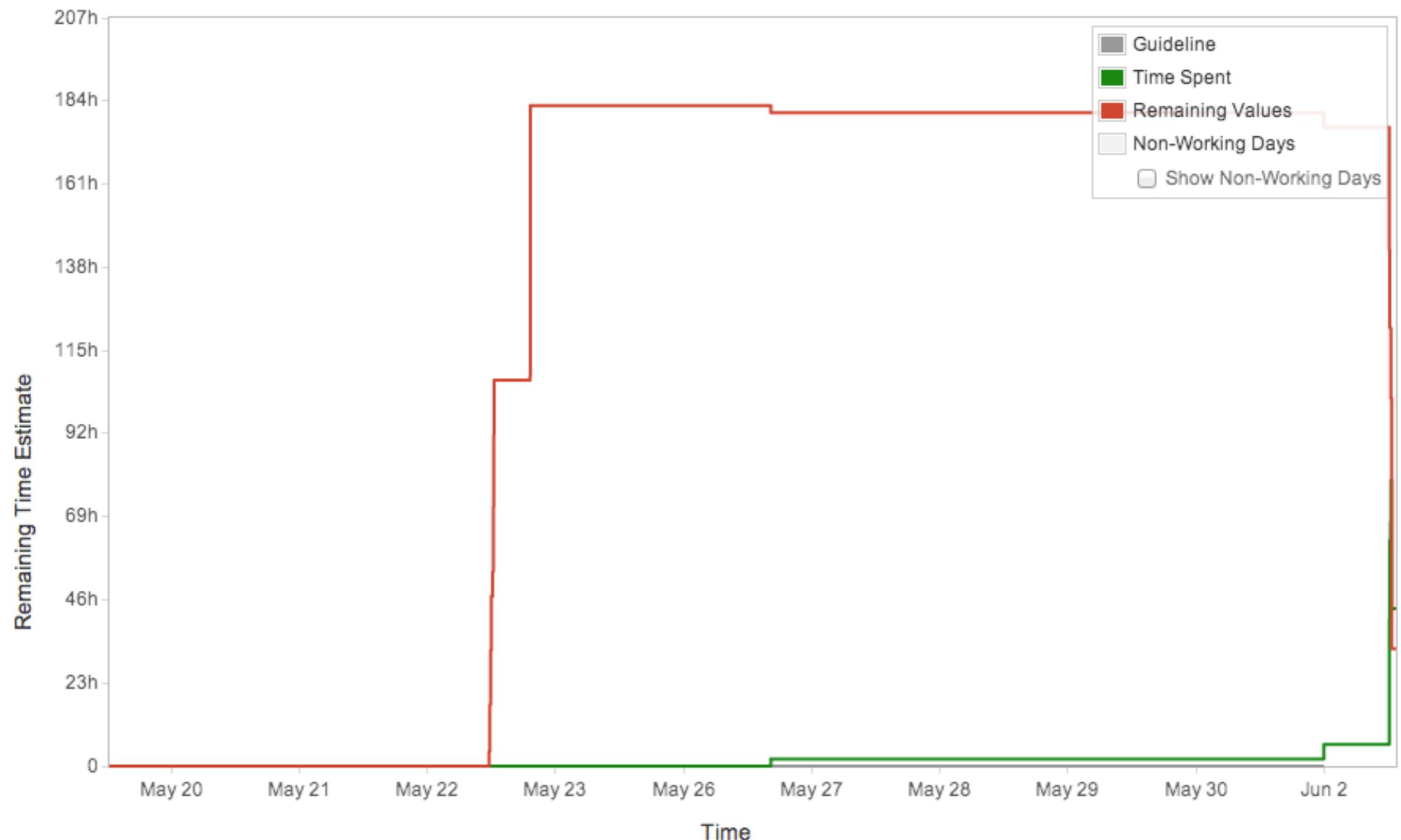
Report by:	Name Surname		Period:	March 31-Apr 6	
Task	Planned time	Tracked time	Completed	Needs review?	URL of results
Look Document For Segments	1:30	1:20	No	No	
Look Document For Canny Edges	1:30	1:10	No	No	
Mentor Personal Meeting	0:30	0:27	Yes	No	
Revise Architecture View	0:30	0:42	Yes	Yes	https://drive.google.com/?usp
Look Document For Canny Edges	1:00	1:10	No	No	
Write Experiment Plan	1:30	1:20	Yes	Yes	https://drive.google.com/?usp
Look MOM With David	1:00	0:48	Yes	No	
Read ACDM paper	1:00	1:04	Yes	No	
Totals:	8:30	8:01			

Design Phase: Tracking



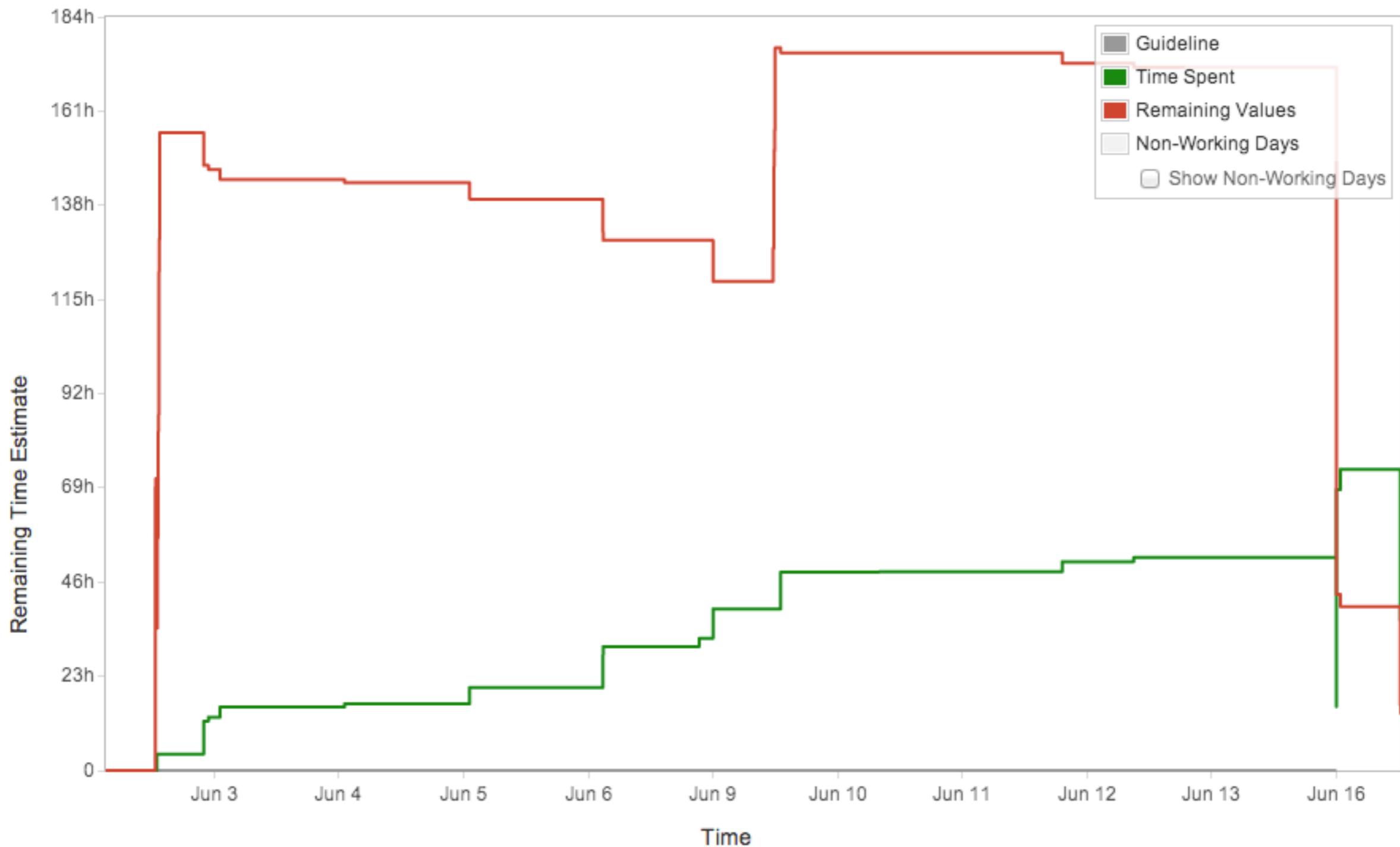
Construction Phase: Tracking

Burndown Chart Sprint 1 ▾



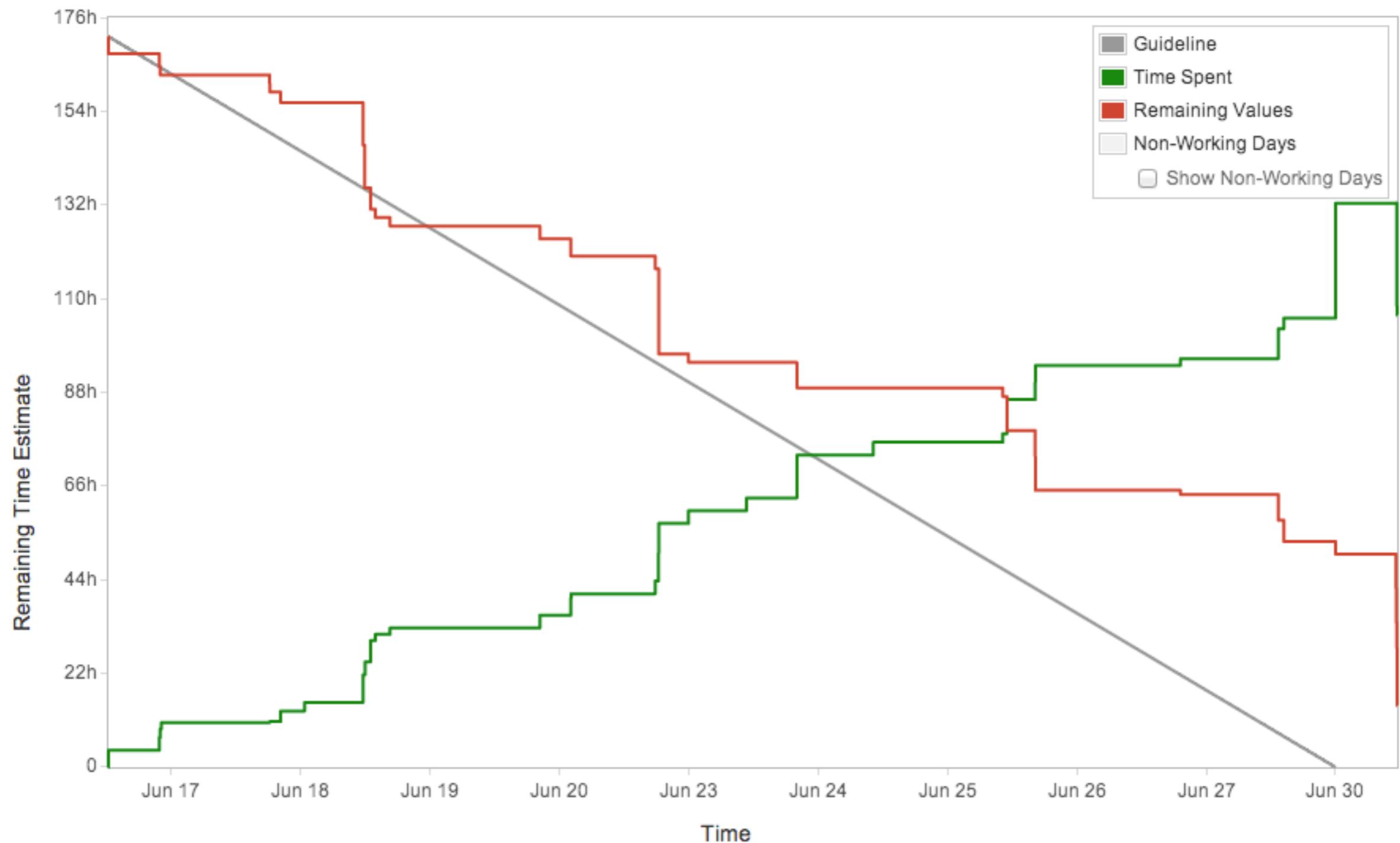
Construction Phase: Tracking

Burndown Chart Sprint 2 ▾



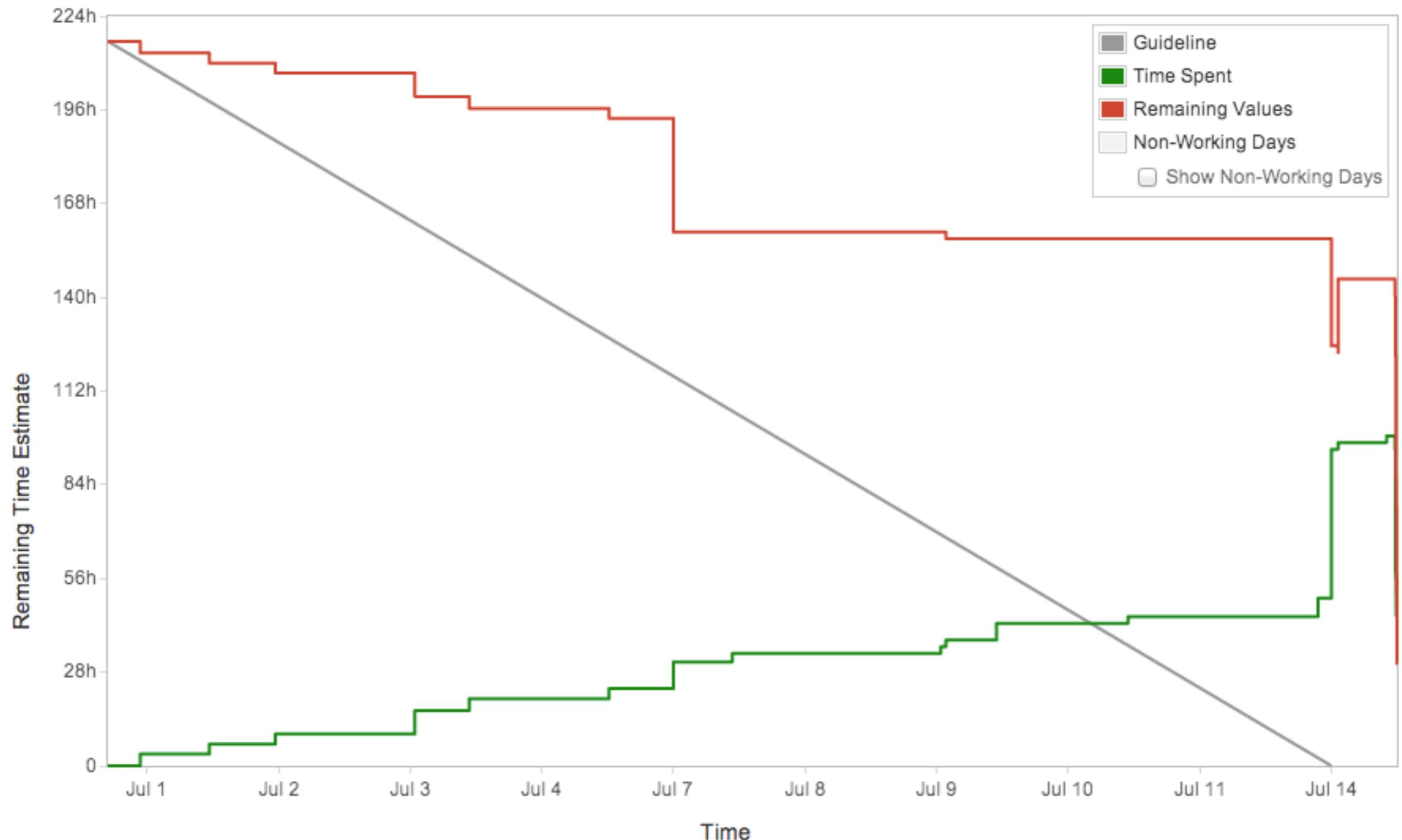
Construction Phase: Tracking

Burndown Chart Sprint 3 ▾



Construction Phase: Tracking

Burndown Chart Sprint 4 ▾



Design Roles

ACDM Roles:

Zakhar: Project Manager + Production Engineer

Pranav: Chief Architect + Production Engineer

Ulyana: Requirements/UI-UX + Production Engineer

Mengyan: Chief Scientist + Production Engineer

Why Scrum?

Question	RUP	MSS	TSP	XP	Scrum
Team Size	1	1	3	3	3
Total Developers	2	2	2	2	2
Product Size and Complexity	2	2	2	2	2
Competent and Experienced Developers	3	3	1	3	3
Level of Hacker Sentiment	2	2	1	2	2
Management Style	2	3	1	3	3
Organization-Wide Processes	2	2	2	3	3
New Process Adoption	2	2	2	2	2
Type of Product	2	2	2	2	2
Requirements Stability	2	2	2	3	3
Requirements Traceability	1	2	1	3	3
Totals	21	23	19	28	28

Scrum Artifacts and Activities

Artifact: Product Backlog

Activity: Product Backlog Refinement

Activity: Sprint Planning

Determine what work will be completed in the Sprint.

Determine how the work will be accomplished.

Artifact: Sprint Backlog

Artifact: Product Increment

Agreement: Definition of Done

Activity: Daily Scrum

Activity: Sprint Review (product)

Activity: Sprint Retrospective (process)

Roles

Product Owner - customer, distributed role

Scrum Master, Project manager - Ulyana

Development Team Member - Pranav, Zakhar,
Mengyan

Additional Roles:

Agile Architect - Pranav

Quality Manager - Zakhar

Technical lead - Zakhar

Chief Scientist - Mengyan

UX Expert + Product Manager - Ulyana

Architecture: Business Context

- Client – David Palmer (Robotics Entrepreneur)
- Client’s Objective:
- Create MVP (minimum viable product)
- Showcase the product to venture capitalists
- Markets to be Captured – Universities (alumni), Museums
- Existing System – Reflect Web Application (under development)

Architecture: Business Context

Stakeholders

Client - David Palmer

Team Reflect Mobile

Mentor - Matt Bass

Third party subcontractor - FalseFit

Subscribers:

University Alumni Organizations

Museums and History societies

High Level Functional Requirements

1. Create the mobile version of Reflect
2. Object boundary detection and categorization
3. Audio recording and uploading
4. Alerts on activity relevant to the users

Architecture: Business Constraints

- **Company standards** - New Mobile application should adhere to the company vision and be consistent in terms of core functionality, look-and-feel and data.
- **Schedule constraint** - The team of 4 team members have to spend 12 hours per week during the spring semester (14 weeks) and 24 hours per week during the summer semester (11 weeks) to deliver the product.
- **Incremental delivery** - During the construction phase the team has to deliver the product incrementally within 2-week iterations.

Architecture: Technical Constraints

- Reflect Mobile application should be deployed on Android device.
- The development must be done within the Android Development Toolkit (ADT) using Java programming language
- Mobile application should use the API of the existing backend services provided by Reflect (legacy database)

Architecture: Quality Attributes

1) **Performance** (in terms of):

- a) **responsiveness of the UI** (Longer computation task does not block the application and runs in the background.)
- b) **efficiency of boundary detection algorithm**
- c) **response time of uploading a file/image** - network latency, size of the file (inbuilt image compression to reduce the resolution of the image), low speed internet can delay response time.

2) **Interoperability** - Consistency in the data between the mobile and web application. **Example** - Photo uploaded from the mobile application should be reflected in the web application and vice versa for a given user.

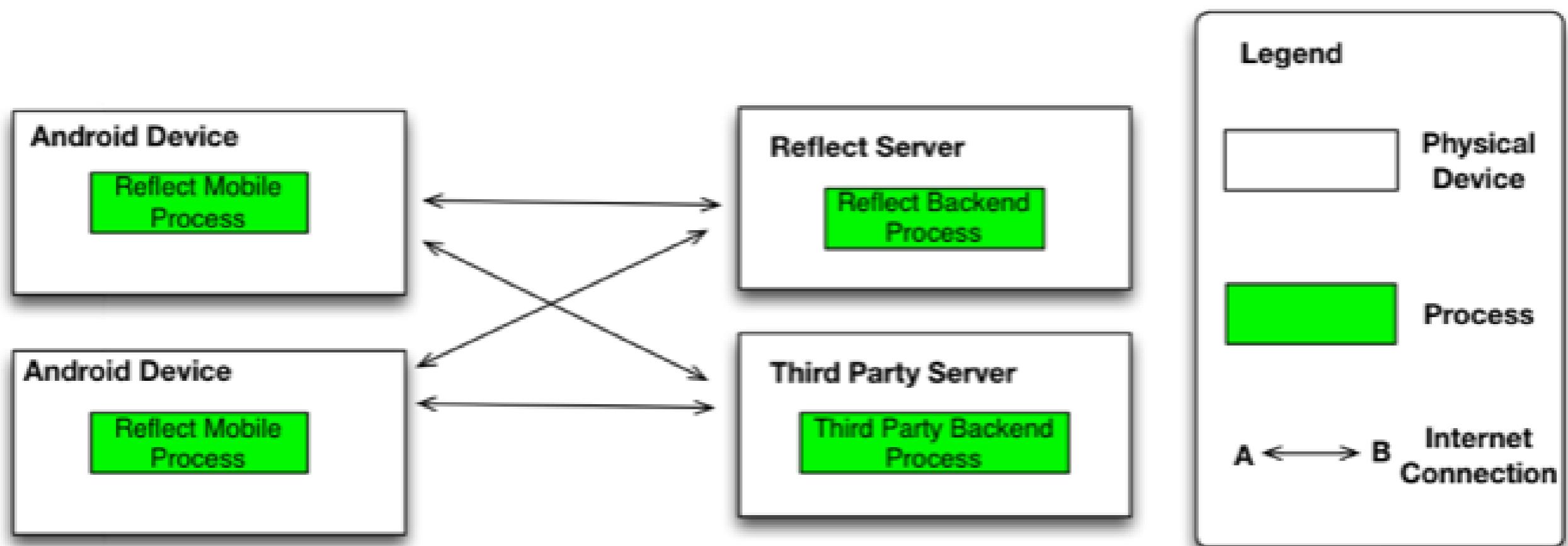
Architecture: QA Performance

Quality attribute: Performance	ID: QAS4
Priority	Very high (Rank 1)
Difficulty	Low
Stakeholders affected:	End user
Reason	The mobile version of Reflect should provide competitive user experience in terms of performance in comparison to performance of modern social media mobile applications.
Stimulus	User of the application performs an activity within the application: user logs in; user navigates to network / community / moment / photo; user adds stories / audios / comments to the photo; user classifies the object with selected boundary
System response	The system provides functionally correct response. Longer computation task does not block the application and runs in the background. Examples of background tasks: file uploading, background image processing.
Response measure(s)	The application responds back within 100 ms. Response measure for

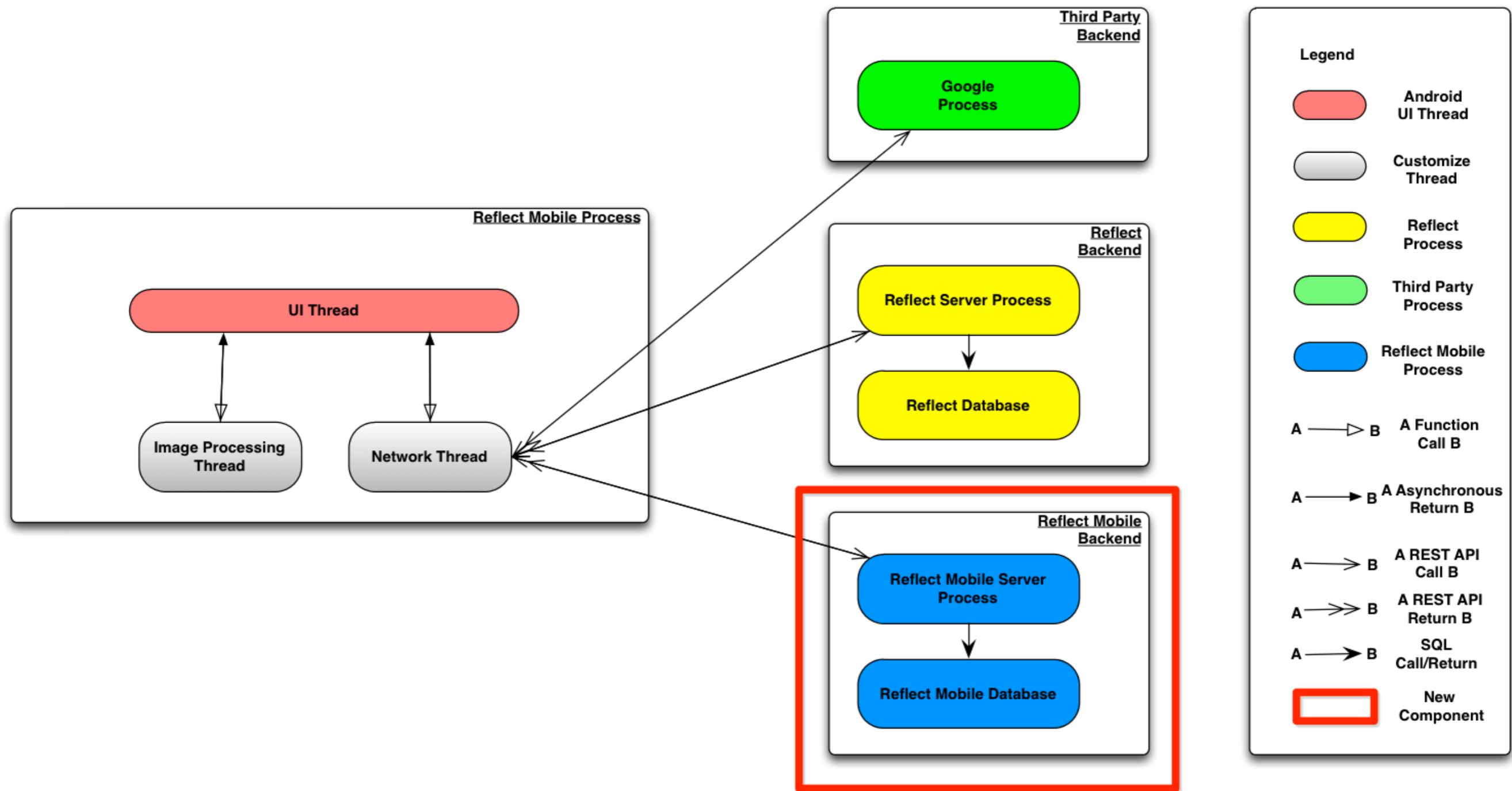
Architecture: QA Performance

Quality attribute: Performance	ID: QAS1
Priority	High (Rank 2)
Difficulty	High
Stakeholders affected:	End User
Reason	When the users choose smart mode (tapping the screen for a long time), the boundary detection algorithm may take a long time to calculate all boundaries in the image.
Stimulus	The user triggers start of the boundary detection algorithm by long-tapping the region of the photo
System response	The system generates a boundary of the region around the center of tap or reports that the boundary could not be detected
Response measure(s)	The system must be able to generate the response within 5 seconds.

Physical Perspective, Allocation View



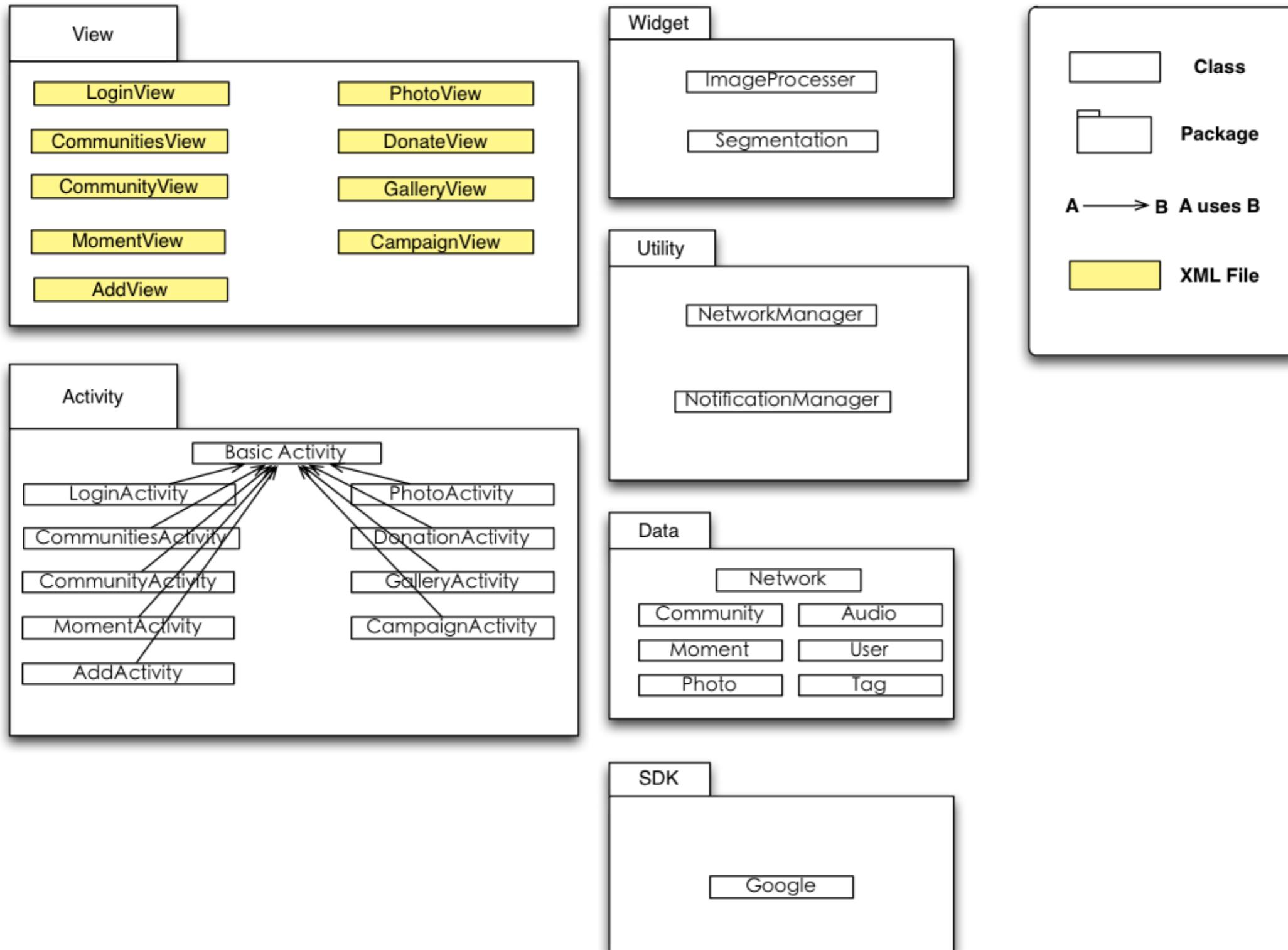
Dynamic Perspective, C&C View



Dynamic Different Processes/Threads, C&C View

1. Reflect Mobile Process
2. UI thread – Renders the UI interface, user input and perform general computations
3. File Process Thread – Writes data to the file system of the mobile device
4. Network Thread – Retrieves data from the network
5. Alert Thread – Sending alerts to the user through the Android framework

Static Perspective, Module View



Top Design Decisions

Allocating responsibility for image processing to the client
Image Processing Task runs on the client device which promotes:

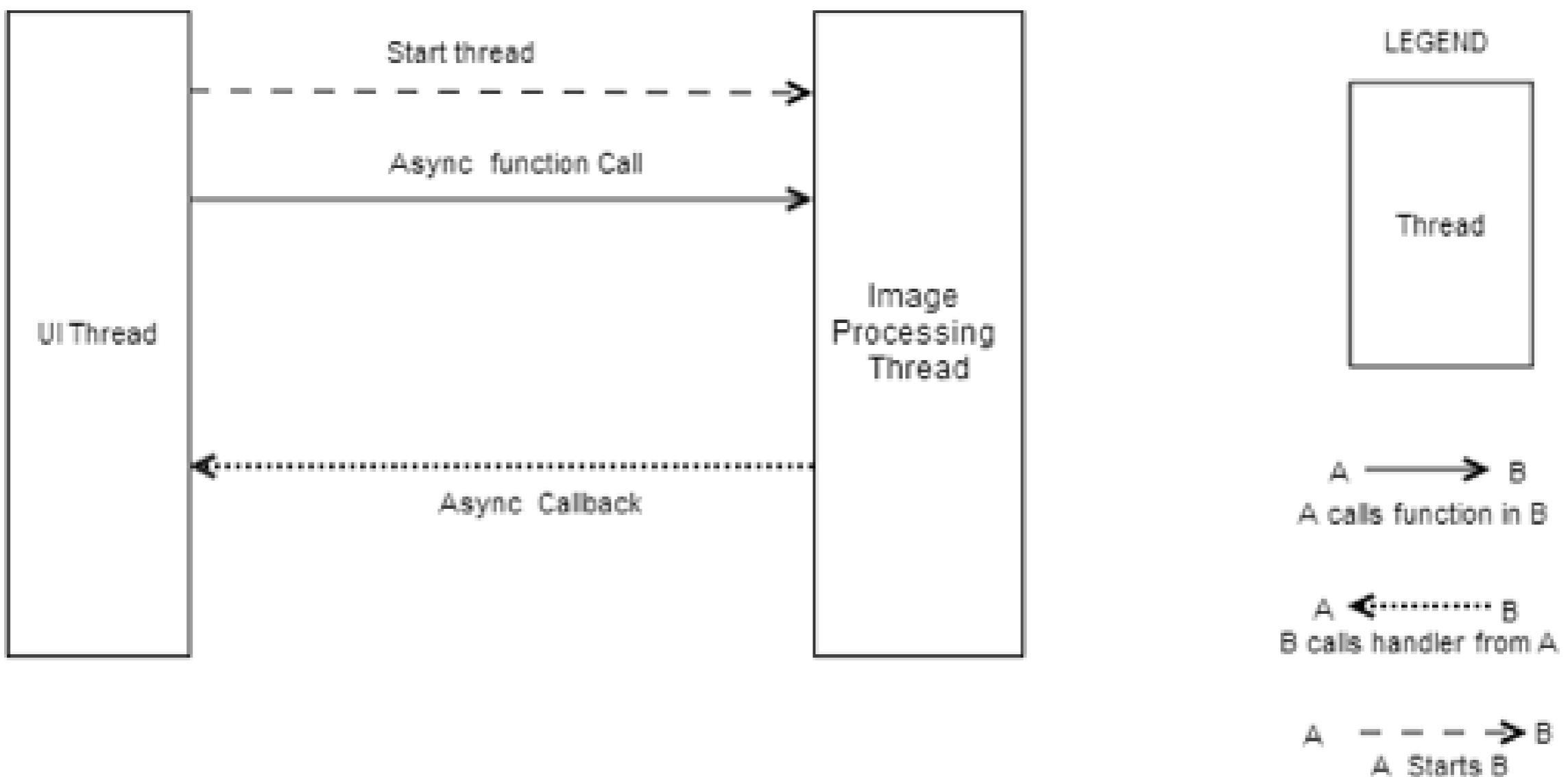
- Performance (verified by experiments)
- Network-connectivity independence (No network reliability, network performance issues)
- Reduced Development Effort

Asynchronous Model of communication

UI thread makes async function call to the Network Manager and Image Processor which in turn call async callbacks of the UI thread.

Promotes responsiveness (QA - Performance)

Sequence Diagram - Asynchronous Thread Communication



Alternatives & Tradeoffs

Alternative 1:

Shifting the image processing functionality on the server (middleware)

- Promotes scalability (addition of more physical servers to support more computationally intensive image processing algorithms)
- Offloads computationally intensive tasks from the mobile device which can promote performance

Tradeoffs – Client Vs Server Side Image Processing:

– Slow & Unreliable Network Connectivity

Due to mobility, the connectivity of the devices can be unstable in terms of both performance (low-bandwidth network) and reliability in case of server side image processing but we get the benefit of network connectivity independence on the client side image rendering.

Alternatives & Tradeoffs

Alternative 2:

Synchronous Model of Communication between threads

Trade Offs:

Inhibits responsiveness of the application (top quality attribute) - hence discarded

Design Phase: Configuration Management

- Agenda template:
 - Formal process for the customer meetings
 - Roles within the meeting: note-taker, time-keeper, facilitator
 - Defined process of storing artifacts from the customer meetings

REWYNDR TEAM MEETING		02/05/2014, 5:45 PM-7:10 PM, SCR 282
Meeting called by	Rewyndr Team	
Type of meeting	Group Meeting with Client	
Facilitator	Zakhar, Pranav	
Note taker	Ulyana	
Timekeeper	Ulyana	
Attendees	Pranav, Zakhar, Ulyana, Mengyan, David Palmer	
Agenda Topics		
1. Rewyndr and Reflect: differences and shared parts. 3 variants of system-to-be 2. Scope exercises 3. Personas, wishes and basic storyboards		
Timestamp	Discussion	Action Items
	1. Rewyndr and Reflect (Zakhar)	
	NB! Requirements change - backend will be reimplemented (Ruby on Rails). Consequences: we cannot use the existing backend. Decision: create 'fungible' system, should be fungible with MVP. Short description of organisational structure:	<ul style="list-style-type: none">1. List of all things that will be done for MVP (Minimum Viable Product) - David2. Update on user stories - David3. Results of Sprints 1-2 - David4. List of tech. questions - team

Construction Phase: Configuration Management

Commit policy.

The procedure goes as follows:

- 1) Team member is assigned a feature
- 2) Team member creates the code for the feature
- 3) Team member tests the solution and fixes the discovered issues
- 4) Team member checks the code for compliance to coding standards
- 5) Team member commits the new changes with descriptive message (definition of done)
- 6) Team member pulls the code from the master branch
- 7) Team member resolves all the merge conflicts if any
- 8) Team member pushes his changes to the master branch
- 9) Assign code reviews to other teammates

Configuration Management: Tools

Google Docs for artifact management

Classify artifact based on different categories

Clean and reorganize monthly

Toggl for time management

Track time for different activities

Source for weekly time report

JIRA for task management

Assign tasks and track status

Source for project tracking and burn-down chart

Github for code management

Currently only for experiments and prototype

Will be reorganized in implementation phase

Quality Plan

ACDM Process Compliance	Quality Activity	Project Activity
	Project Plan Review	Project Planning
	Requirements V&V	Requirement Gathering and Elaboration
	Risk Management	
	Verify Architectural drivers	Discover AD
	Verify Statement of Work	Establish Project Scope
	Analyze Notional Architecture for feasibility	Create Notional Architecture
	Architecture Review	
	Revise Plans	Experiments Planning
	Review of experiments results	Experiments Execution

Quality Plan

Scrum Process Compliance	Design Inspections	Production Planning (Detailed Architectural Design)
	<ul style="list-style-type: none">- Code Inspections<ul style="list-style-type: none">- Testing- Demonstrations- Reflection	<p>Scrum #N</p> <ul style="list-style-type: none">- Planning- Implementation- Reflection
	<ul style="list-style-type: none">- User Acceptance Activities<ul style="list-style-type: none">- Usability Testing	User Acceptance

Notional Architecture Analysis

Artifact: Architectural Documents - Static View Diagram, Dynamic View Diagram, Physical View Diagram, Architectural Report

Planned Time: This activity should be executed in the end of the ACDM phase “Create Notional Architecture” . It needs to be held during two days, 4 hours per day.

People: All team members

Tasks:

- Correctness - Verify that tradeoffs made by team correspond to prioritized quality attribute scenarios
- Consistency - Verify that different views within the Notional Architecture are consistent with each other.
- Completeness - Verify that all of the tradeoffs made for the current partitioning of the system have been documented. Verify that all of the high-level functional requirements have been addressed in the Notional Architecture.
- Readability- Verify that the Notional Architecture documentation is understandable and unambiguous.

Notional Architecture Analysis

Entry and exit criteria:

Entry Criteria: Documented version of the following artifacts: architectural drivers specification, Notional Architecture,

Exit Criteria: Refined version of the architectural document.

Required Effort: 32 man-hours

Required Tool: Microsoft Word or other text editor capable of modifying document format.

Required Training: No

Measurement: Tradeoffs, architectural drivers and notional architecture are documented and follow-up changes made to architectural document

Testing

Testing UI Checklist

Verify all alignments and sizes of all elements. Alignment and sizes should be proper/same as defined in wireframe.

Spell check all text in all screens. There should not be any errors.

Verify that Font sizes are consistent. Font sizes for headers, input fields, all labels should be consistent, refer to wireframes. Ensure that the screen font family matches from screen to screen.

Verify that any kind of text is cut off.

Verify that any kind of graphics is not blurred, distorted.

Verify that there is visual feedback when response to any action takes more than 3 seconds.
If the screen takes more than 5 seconds to display the results/page, it should contain a progress bar so that the user understands the processing is continuing.

Verify that each screen should be visible for the time necessary to comfortably read all its information by moving between screens of an application. Each screen must be visible for the time necessary to comfortably read all its information.

Verify that error messages in the Application are clearly understandable. Ensure that error messages are informative, grammatically correct, and not condescending.

Verify that error messages clearly explain to a user the nature of the problem, and indicate what action needs to be taken (where appropriate).

Verify that any function selected in the Application should start within 3 seconds.

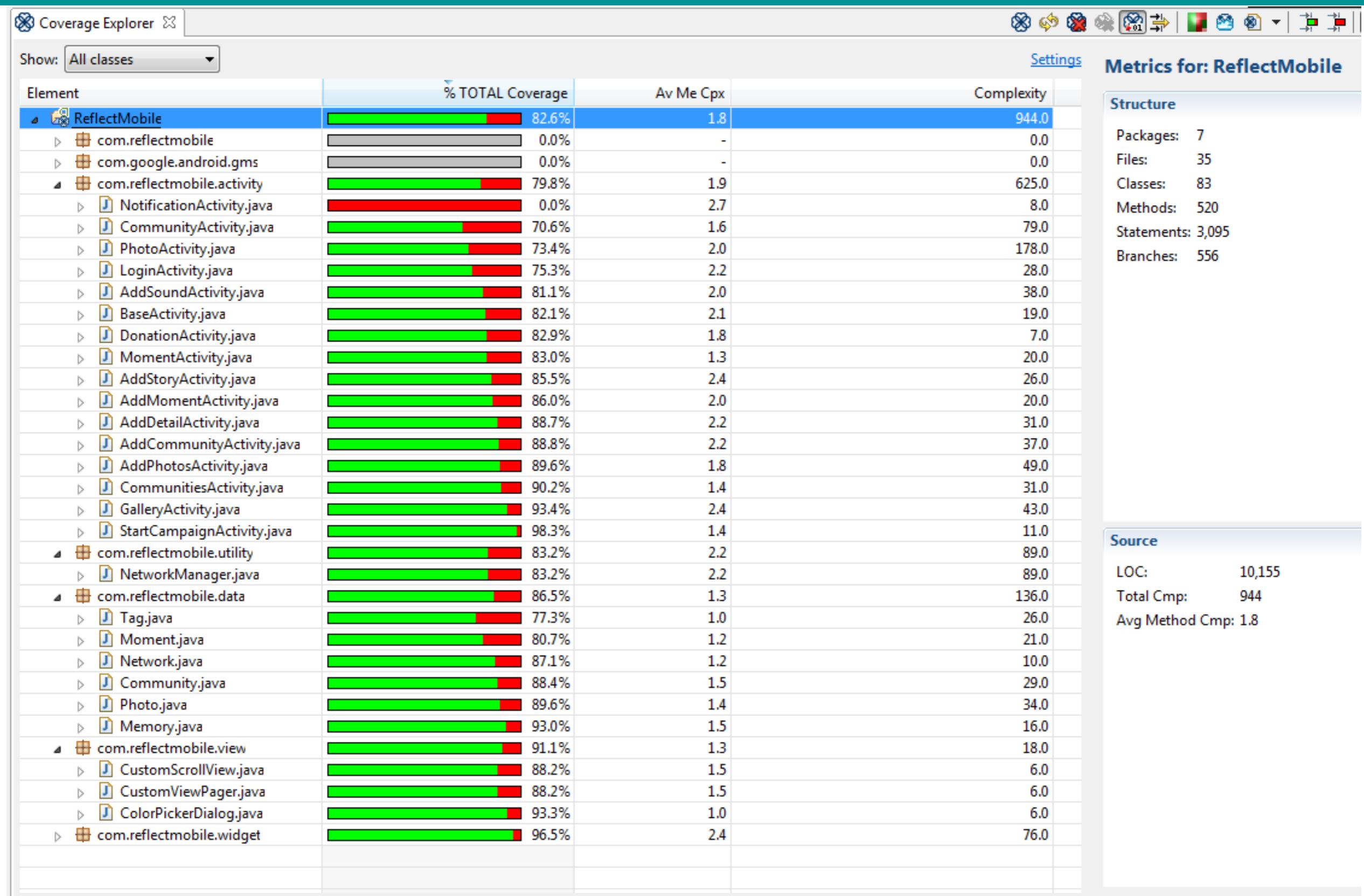
Verify that If the screen contains text boxes that allow data entry, ensure that the width of data entered does not exceed the width of the field.

Testing

Script-based Testing|

Test Case	Test "viewing"			
	Step Description	Expected Result	Actual Result (if different from expected)	Successful/Failed
1	<i>Log in via Google account</i>	No errors, logged in	No errors, logged in	Successful
2	<i>View communities</i>	No errors. Should be able to list all communities, scroll if more than 4 communities.	No errors. Should be able to list all communities, scroll if more than 4 communities.	Successful
3	<i>Select community</i>	No errors. Selecting empty community should lead to "add new photo" activity.	No errors. Selecting empty community should lead to "add new photo" activity.	Successful
4	<i>View moments</i>	No errors. Should be able to list all moments, scroll if more than fit one screen.	No errors. Should be able to list all moments, scroll if more than fit one screen.	Successful
5	<i>Filter the moments by year. Select specific year.</i>	No errors. After the year is selected only moments from this year are shown.	No errors. After the year is selected only moments from this year are shown.	Successful
6	<i>Select moment</i>	No errors. Selecting "add new photo" icon should lead to "add new photo" activity.	No errors. Selecting "add new photo" icon should lead to "add new photo" activity.	Successful
7	<i>View photos</i>	No errors. Should be able to list thumbnails, scroll if more than fit one screen.	No errors. Should be able to list thumbnails, scroll if more than fit one screen.	Successful
8	<i>Tap "tag" button</i>	No errors. Leads to the "tag mode"	No errors. Leads to the "tag mode"	Successful
9	<i>View tags (select one by one), view memories for the tag</i>	All existing tagged regions are shown. Faces are detected and tagged regions are shown for them. When the tag is selected, all memories are listed for this tag.	All existing tagged regions are shown. Faces are detected and tagged regions are shown for them. When the tag is selected, all memories are listed for this tag.	Successful
			Test Case Status	Successful/Failed

Construction Phase – Quality Mgmt



Risk Management

Source of Risk	Condition	Consequence	Mitigation	Probability	Impact	Status
Development Process (Team Time Resources)	Team members tend to take lot of electives	Team members might be overloaded with electives. This can affect deadlines	Proper planning, buffer period, distribute project resources accordingly	High	High	Problem (Open)
	Team members (resources) tend to search for jobs in this period.	Resource can become unavailable (Interviews / trips / any other errands). This can affect deadlines	Planning to compensate hours devoted to development in case of interviews. Policy for time compensation.	High	High	Problem (Open)
Design	Android components difficult/ impossible to implement	Project might lose value, client may not be satisfied	Scope Exercises, early experimentation	Medium	High	Mitigated
	Due to limited time, team members tend to spend most of the time in development which can	This might lead to a situation where the features are not working as expected and thus the project will not have any value	Thorough quality plan, rigorous testing required	Medium	High	

Risk Management

	affect the testing of the features.					
Scope & Estimates	Team members could not estimate appropriately about the timeframe to prepare the framework for the application on which object boundary detection is dependent	Wrong estimation might lead to a situation where there is not enough time for object boundary detection feature to implement. Project will have no value as this is the core feature of the application	Prioritize feature development and estimate the cost of developing this feature Vs other secondary features	Low	High	Mitigated
Process Quality (Configuration Management)	Improper process for code integration (specially dependent features) results in unwanted merge conflicts	Can result in undesirable code loss, wastage of time in code retrieval	Define a policy for code integration and handling dependent features merge requests (who commits when)	Low	Medium	Partially mitigated, still a risk

Risk Management

Product Quality	Team members don't unit test their code and don't handle boundary cases	Application might have bugs in which case product value will diminish, customer & stakeholders will be unhappy	Incorporate proper quality plan into the process (Exploratory testing, integration testing and basic assertions)	Medium	High	Partially mitigated, but still a problem
	Team members don't adhere strictly to the coding conventions	Code quality might become low which leads to low product quality, low maintainability	Follow established coding standards, enforce code reviews	High	Medium	Mitigated
Requirements	Customer changes requirements	Can affect development process, compromise code quality, product with very less value in the worst case	Negotiate proper scope of the features to be implemented (must-haves)	Low	High	Problem, but resolution is in place (no nice-to-haves)
Development Environment	Android updates SDK	Affects the UI , product quality will suffer, implementation will have to be re-adjusted which can affect deadlines	No concrete mitigation strategy possible except for not updating the SDK	Medium	Very Low	New Risk

Risk Management

Communication with Truefit	Truefit is not going to implement any request made by our team	That means that either we have to remove features from the scope, or widen our scope by creating custom backend	Plan in advance for possible rejections from the Truefit side. Understand the limitation that the backend can't change under any circumstances.	High	High	Problem, but we were prepared for that
----------------------------	--	---	---	------	------	--

Design Phase – Risk Mgmt.

Source of Risk	Condition	Consequence	Mitigation	Impact	Probability	Status
Implementation/ Technology	Algorithms do not meet QA		Start experiments as soon as possible	High	High	Closed
Existing product, communication with subcontractors			Track backend team's progress, visit Sprint meetings, conduct integration meetings, define points of synchronization	High	High	Open
Time	Team members overloaded with core courses and electives		Create weekly work sessions	High	High	Open