ShopAdmin

Control your expenses with receipt scanner

Content

Kateryna Pryshchepa, Iuliia Guk

ShopAdmin

Control expenses app with receipt scanner

Vision

Whom is it for?

students or young family.

Functionality:

- Manually enter your outgoing transactions
- Scanners that auto-extract receipt information such as shop names, data and time, amount

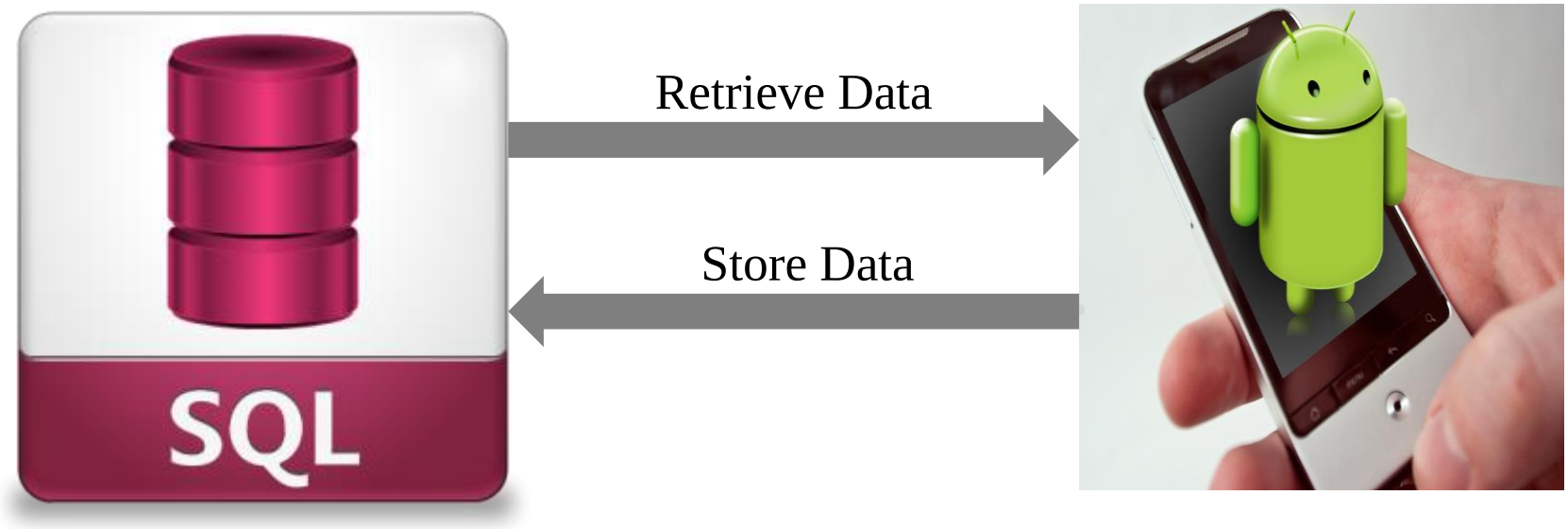
What problems does it solve?

Save time for counting expenses.

What alternatives are available?

- Cashbook Expense Tracker
- Expense Manager
- Expensify

Architecture



Model	Controller	View
Store required info: shop name, date and time, amount.	Number & text recognition OCR (optical character recognition)	•Display receipt information •Sort and categorize expenses •Summarize expenses

Challenge

- Using Android for the first time
- Implementing Tesseract Library OCR engine and Leptonica Image Processing Library

Example:

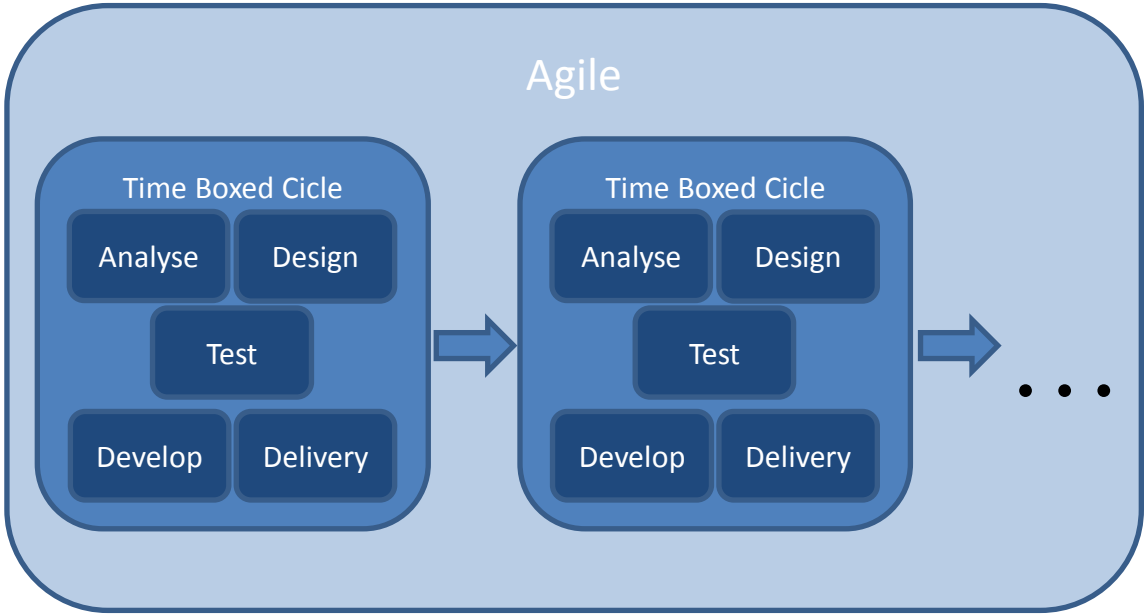
Receipt	Amount
ALDI	19.86
PENNY	15.31
REWE	25.35
Total:	60.52

Content

- In general the development of ShopAdmin will take place in an agile way
- Mainly differentiation between being agile and a classical approach
  - time boxed approach
- But furthermore agile has its merits and principles

Merits

- Commitment
- Focus
- Openness
- Respect
- Courage
- Simplicity
- Communication
- Feedback

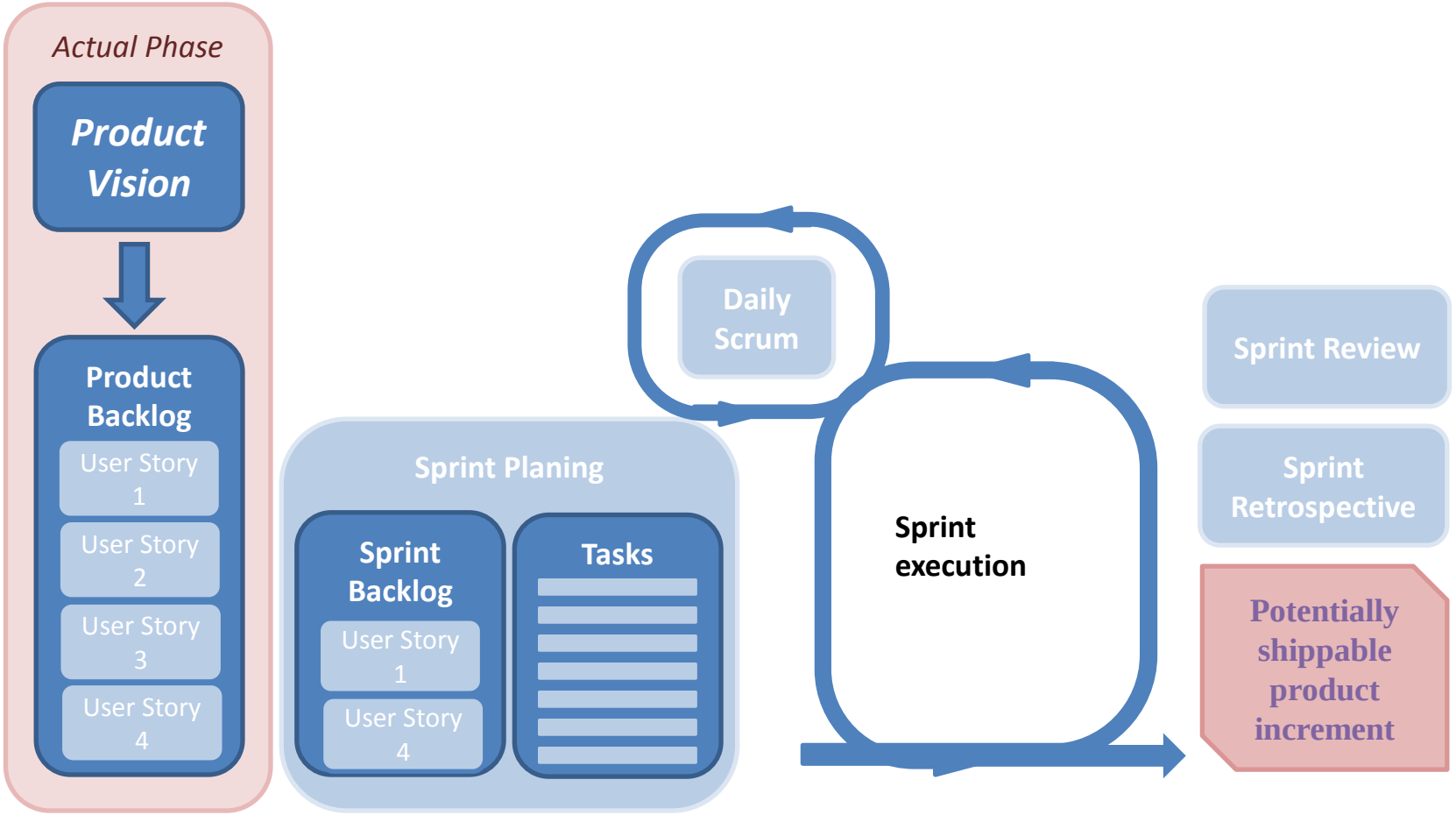


Principles

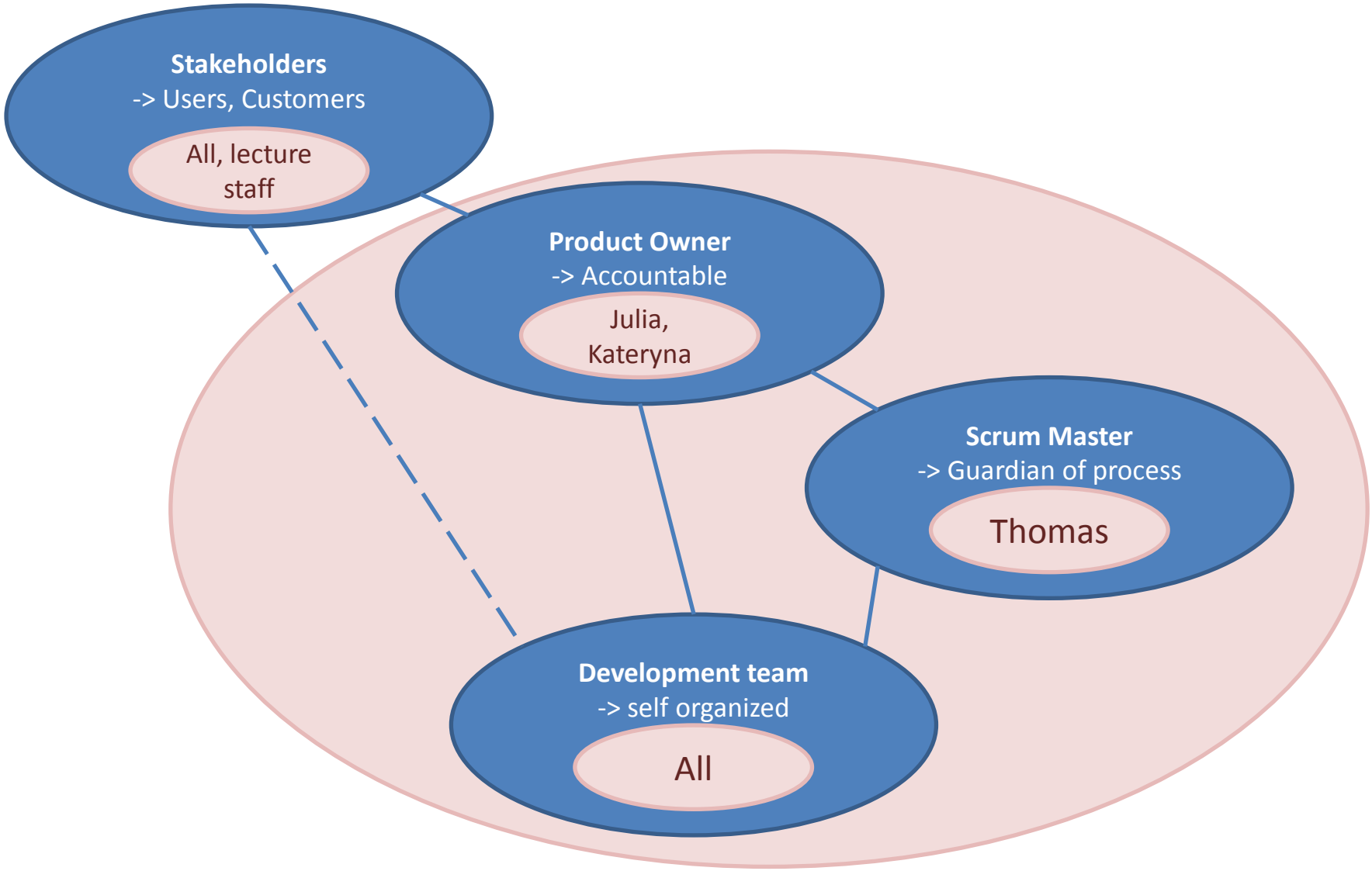
1. Satisfy the customer
2. Welcome changes
3. Frequent deliveries
4. Cross-functional
5. Support and trust
6. Direct communication
7. Working software
8. Sustainable speed
9. Ambition for technical excellence
10. Self organization
11. Review and adapt

Content

• While being agile the development approach of ShopAdmin follows the SCRUM Process



Scrum Team


Stakeholders
-> Users, Customers
All, lecture staff
Product Owner
-> Accountable
Julia, Kateryna
Scrum Master
-> Guardian of process
Thomas
Development team
-> self organized
All

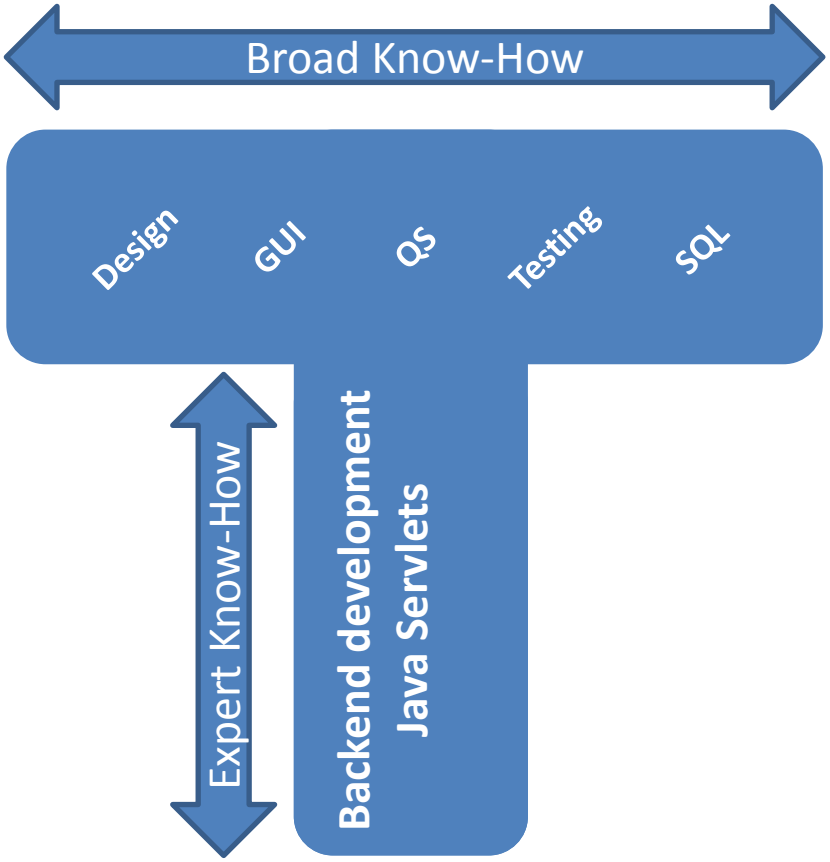
Characteristics:

- Self organizing team following the one-team-approach
- Interdisciplinary
- Close coworkers of the product owner
    - Support regarding product backlog refinement
- Responsible for sprint planning

Attributes:

- 3 – 9 persons
- Ideally T-Shaped members

The T-Shape and you…



Idea:

- Many T-Shapes cover all aspects and everybody has base knowledge in all areas

Content

VISION ESTABLISHMENT *15.11.2015 – 27.11.2015*

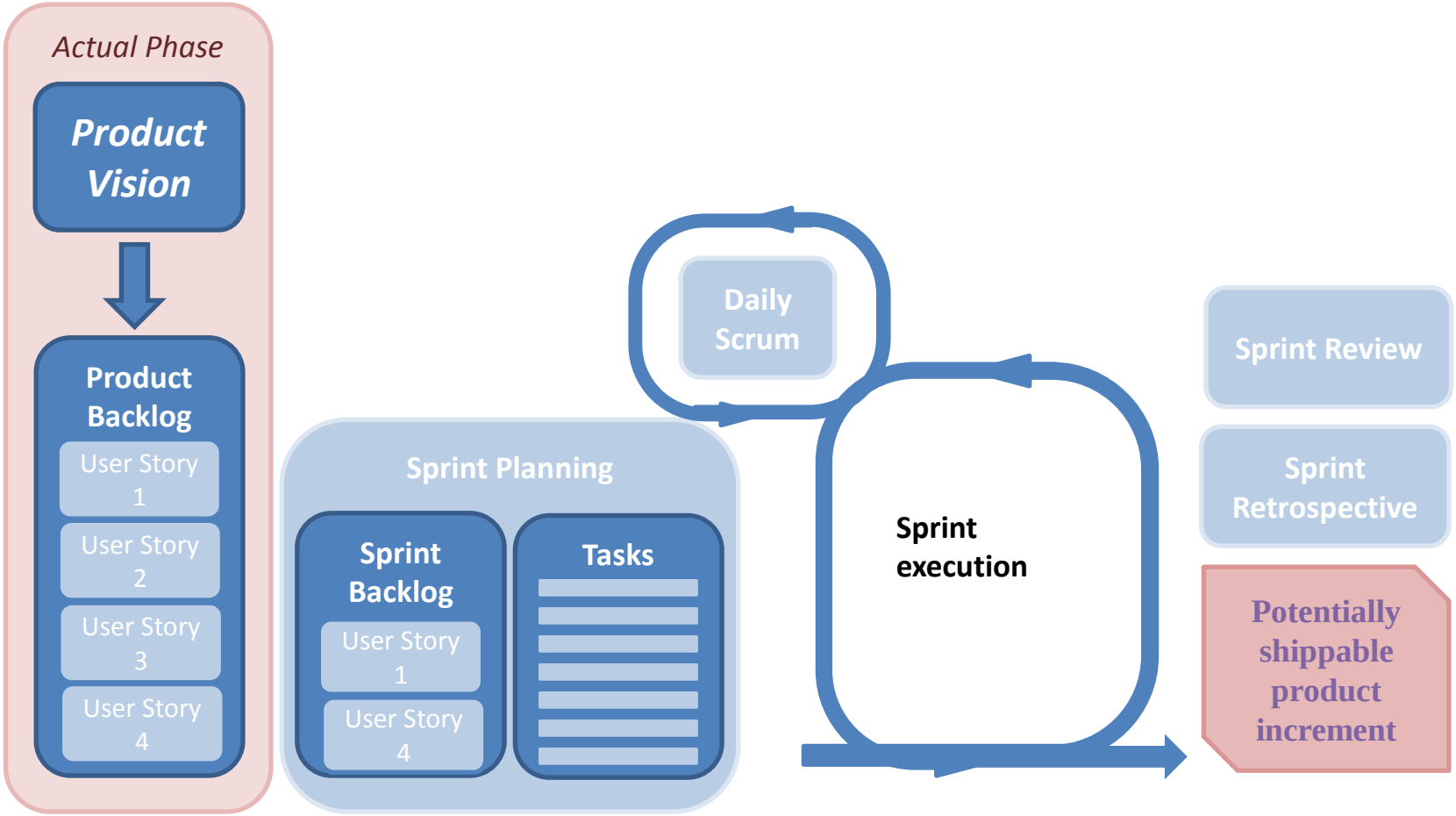
TIMELINE

SPRINT I *27.11.2015 – 08.01.2016*

SPRINT II *09.01.2016 – 22.01.2016*

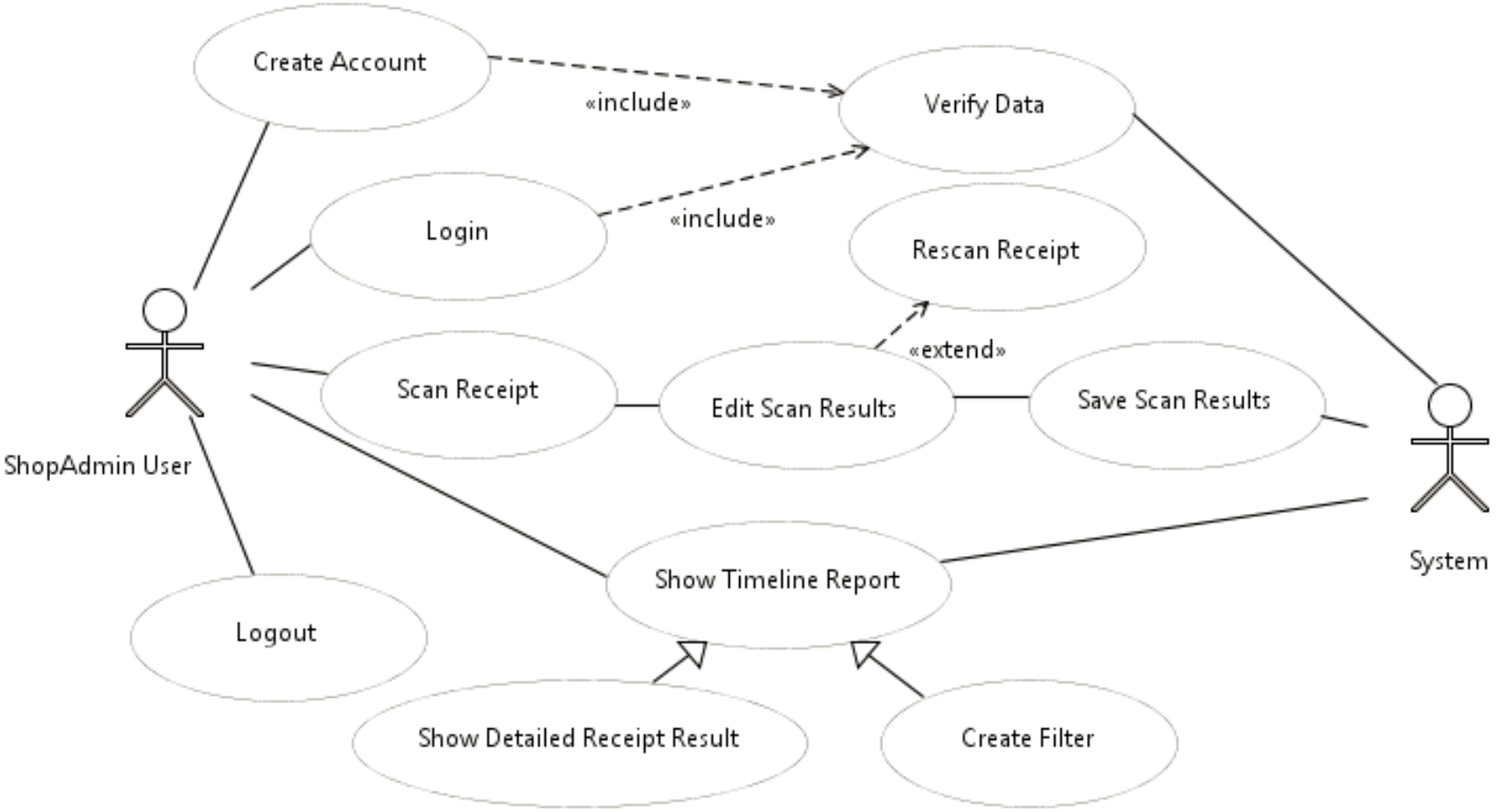
SPRINT III *22.01.2016 – 05.02.2016*

Where are we now?



Starting from the vision of ShopAdmin during the establishment phase following topics have been addressed and solved:

- SetUp product backlog
  - Preparation of user stories
  - Stabilization of vision by UML diagram
  - Concretization of user stories by definition of use cases
- Arrangement of technical infrastructure
  - GitHub: created and tested
  - Development platform: Android SDK available to the team
  - ScrumDesk: Initialized for convenience


Create Account
«include»
Verify Data
Login
«include»
Rescan Receipt
«extend»
Scan Receipt
Edit Scan Results
Save Scan Results
ShopAdmin User
System
Show Timeline Report
Logout
Show Detailed Receipt Result
Create Filter

Definition of done:

1. All tasks have to be completed
2. All implementations have to be commented
3. All tests have to successfully run through
4. Documentation has to be updated
5. Everything has to be pushed to git
6. Scrum user story has to be set to completed

VISION ESTABLISHMENT *15.11.2015 – 27.11.2015*

TIMELINE

SPRINT I *27.11.2015 – 08.01.2016*

SPRINT II *09.01.2016 – 22.01.2016*

SPRINT III *22.01.2016 – 05.02.2016*

- **Timeline 2015:**

Task/KW	*46*	*47*	*48*	*49*	*50*	*51*	*52*	*53*
Product Vision								
Sprint I								
Sprint Planning								
Daily Scrum								
Holiday								

- **Timeline 2016:**

Task/KW	*53*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Holiday									
Sprint I									
Review									
Sprint II									
Planning									
Daily Scrum									
Retrospective									
Sprint III									
Planning									
Daily Scrum									
Retrospective									
Sprint IV									
Planning									
Daily Scrum									
Retrospective									

VISION ESTABLISHMENT *15.11.2015 – 27.11.2015*

TIMELINE

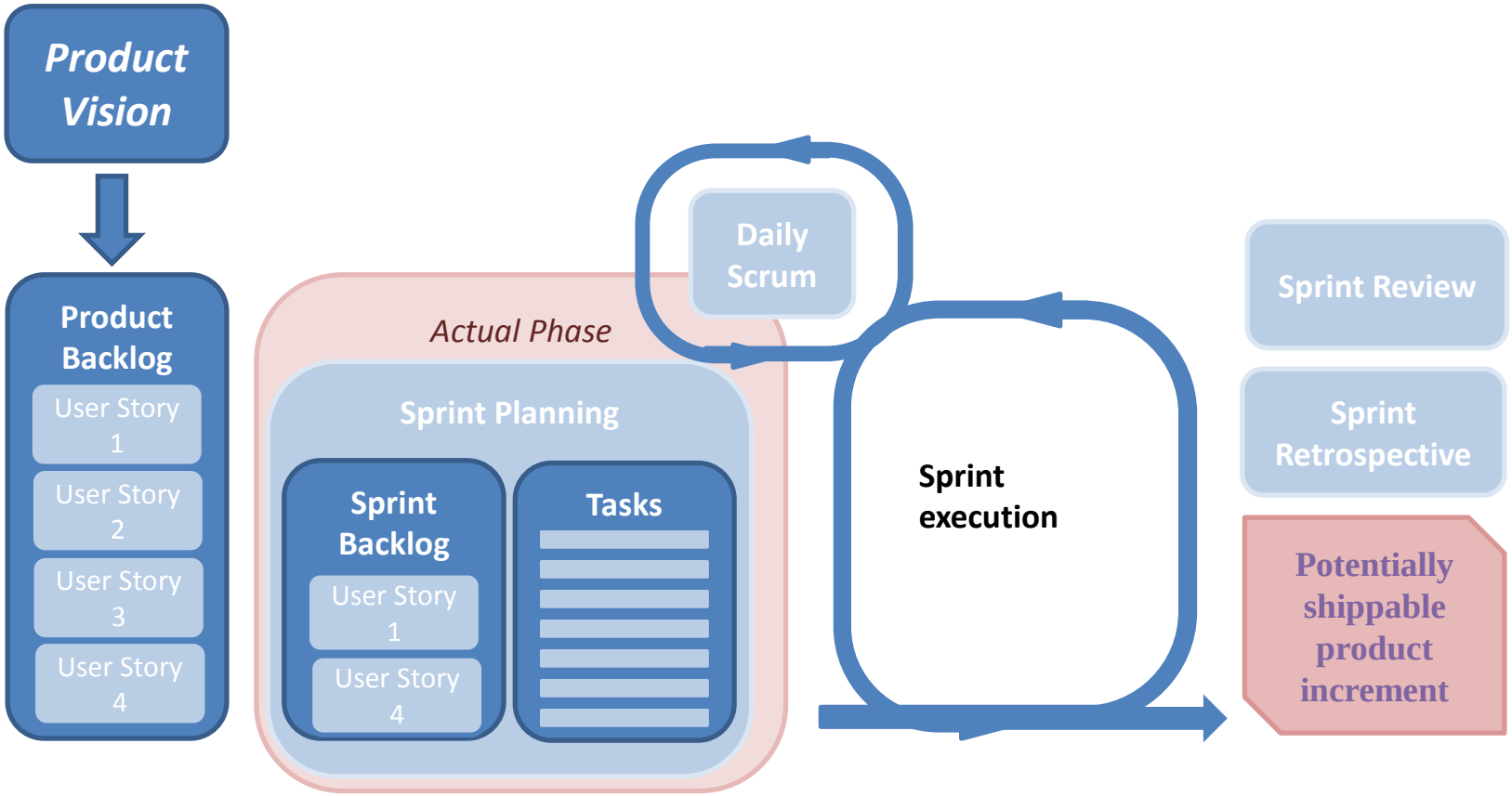
SPRINT I *27.11.2015 – 08.01.2016*

- **PLANNING THE FIRST SPRINT *27.11.2015***
- DAILY SCRUM *04.12.2015*
- DAILY SCRUM *11.12.2015*
- DAILY SCRUM *18.12.2015*

SPRINT II *09.01.2016 – 22.01.2016*

SPRINT III *22.01.2016 – 05.02.2016*

Where are we now?



Purpose of the sprint planning:

- Define the product backlog items which will be developed ***(What?)***
- Plan the development ***(How?)***

Preconditions:

- Product backlog is sufficient cultivated
- Business owner has a clear idea of the sprint goal
- Clarity of the capacity of the development team (resources and tools)

Result:

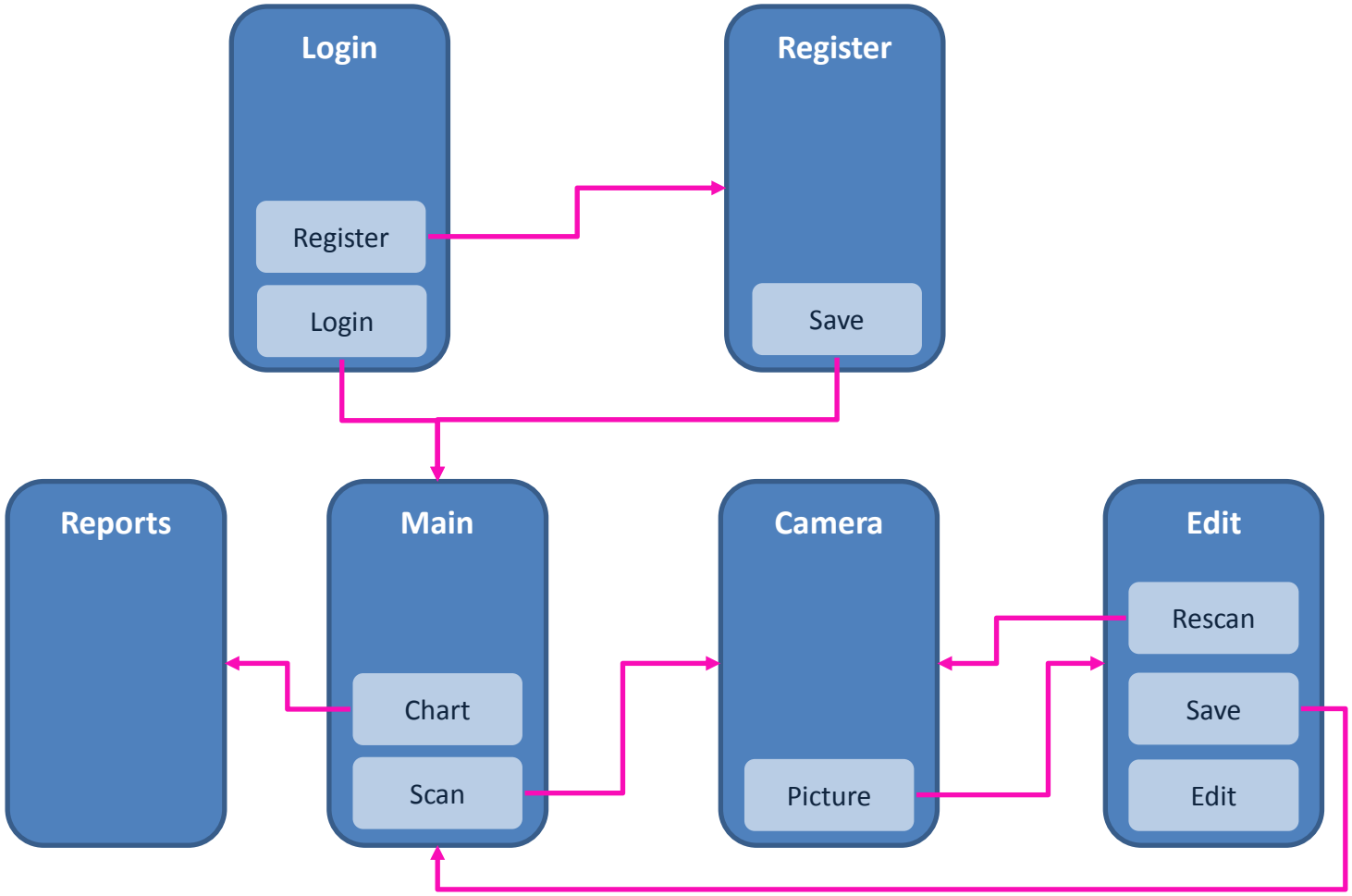
- Formalized Sprint Backlog
- Commitment on sprint goal

The Sprint Goal:

- Main goal for the first sprint is to develop a technical spike for the whole screen flow of the app, to build a sustainable base to grow further in the following sprints.
- Furthermore the UserStories Registration and Login are targeted for implementation.

- Additional benefit in getting used to Android development and strengthening of the SCRUM methodology.

- Scheme of screens:



Vision Establishment *15.11.2015 – 27.11.2015*

Timeline

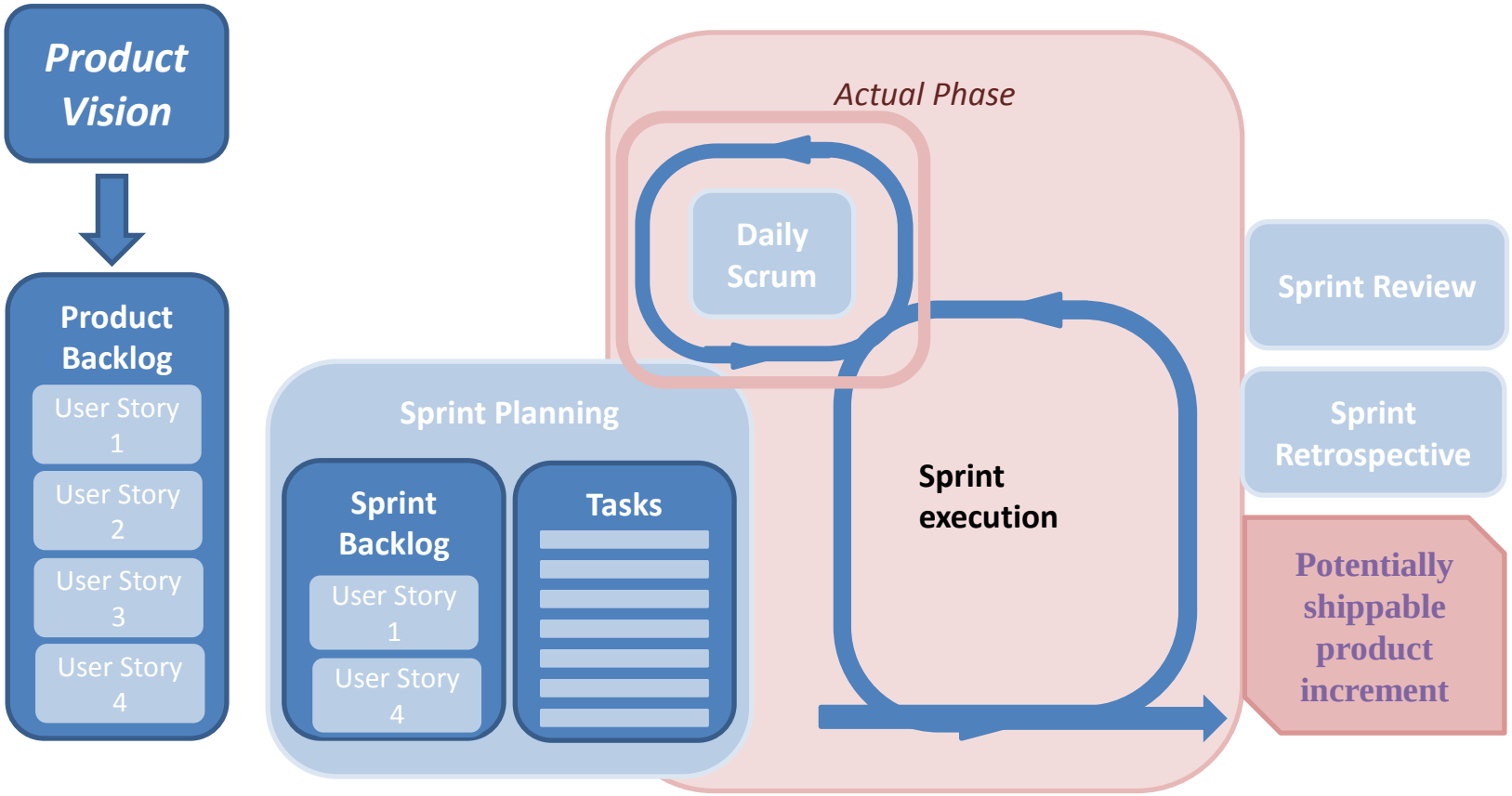
Sprint I *27.11.2015 – 08.01.2016*

- Planning the first Sprint *27.11.2015*
- **Daily Scrum *04.12.2015***
- Daily Scrum *11.12.2015*
- Daily Scrum *18.12.2015*

Sprint II *09.01.2016 – 22.01.2016*

Sprint III *22.01.2016 – 05.02.2016*

Where are we now?



Purpose:

- Synchronization of the development team
- Planning till the next daily scrum ***(Inspect & Adapt)***

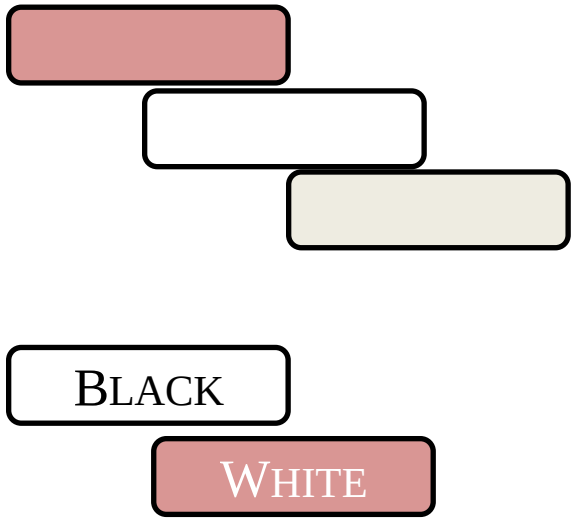
Preconditions:

- Discipline regarding time box, communication, accuracy

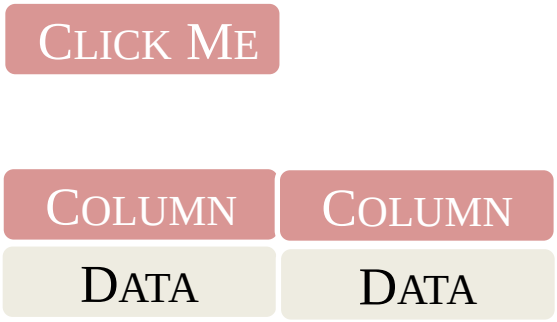
Result:

- Answers on the ***3*** questions by ***all*** participants:
  1. What was achieved since the last daily scrum
  2. What is planed till the next daily scrum
  3. Are there impediments on the way to the sprint goal

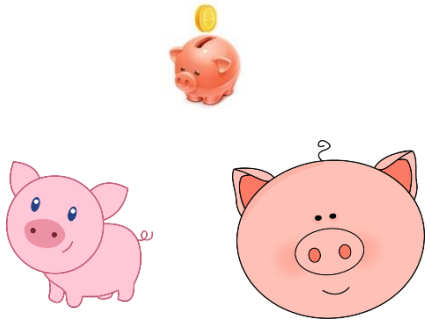
- **MokUp**
  - Corporate Design based on three colors:
    - pink
    - white
    - light grey
    - Font:
      - black
      - white



- **MokUp**
  - Defined GUI-Elements
    - Buttons
    - Tables
      - Header
      - Data
    - 1x Dropdown Box
      - Category on EditView
    - 1x Undefined very fancy element to choose different reports



Impediments:



- **Pigs!** We need more pigs…

- **User Stories**:
  - Dependencies regarding developments.
    - I.e. activity *register* needs to be finished before work on activity *login* can start.
  - Reduce scope of user stories, so that one developer can work individually on one user story.
    - User stories will be recombined via Epics in ScrumDesk
  - Furthermore the impediment adapted to the commonly used android app
    - Technical resolution is based on interfaces

VISION ESTABLISHMENT *15.11.2015 – 27.11.2015*

TIMELINE

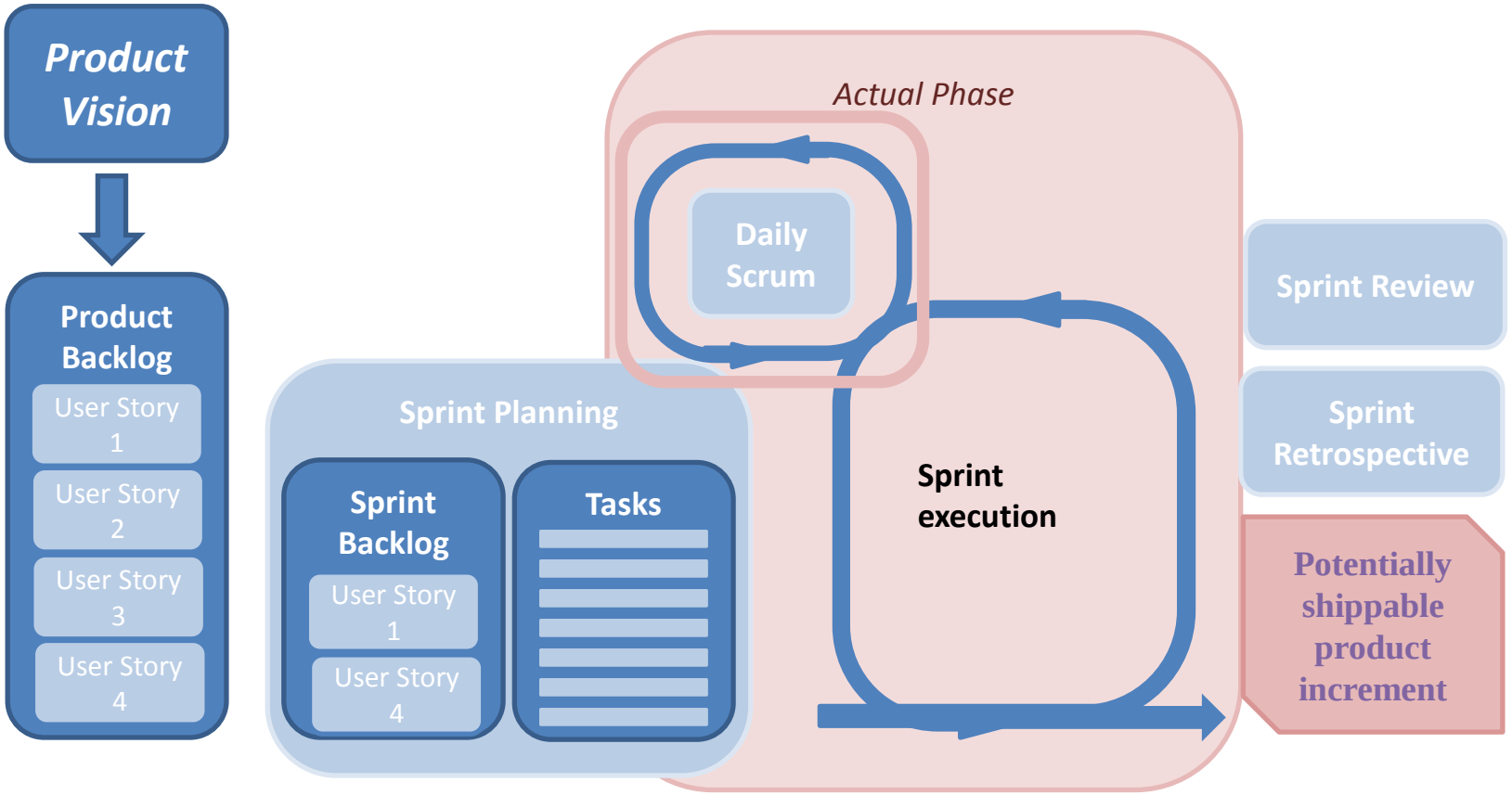
SPRINT I *27.11.2015 – 08.01.2016*

- PLANNING THE FIRST SPRINT *27.11.2015*
- DAILY SCRUM *04.12.2015*
- **DAILY SCRUM *11.12.2015***
- DAILY SCRUM *18.12.2015*

SPRINT II *09.01.2016 – 22.01.2016*

SPRINT III *22.01.2016 – 05.02.2016*

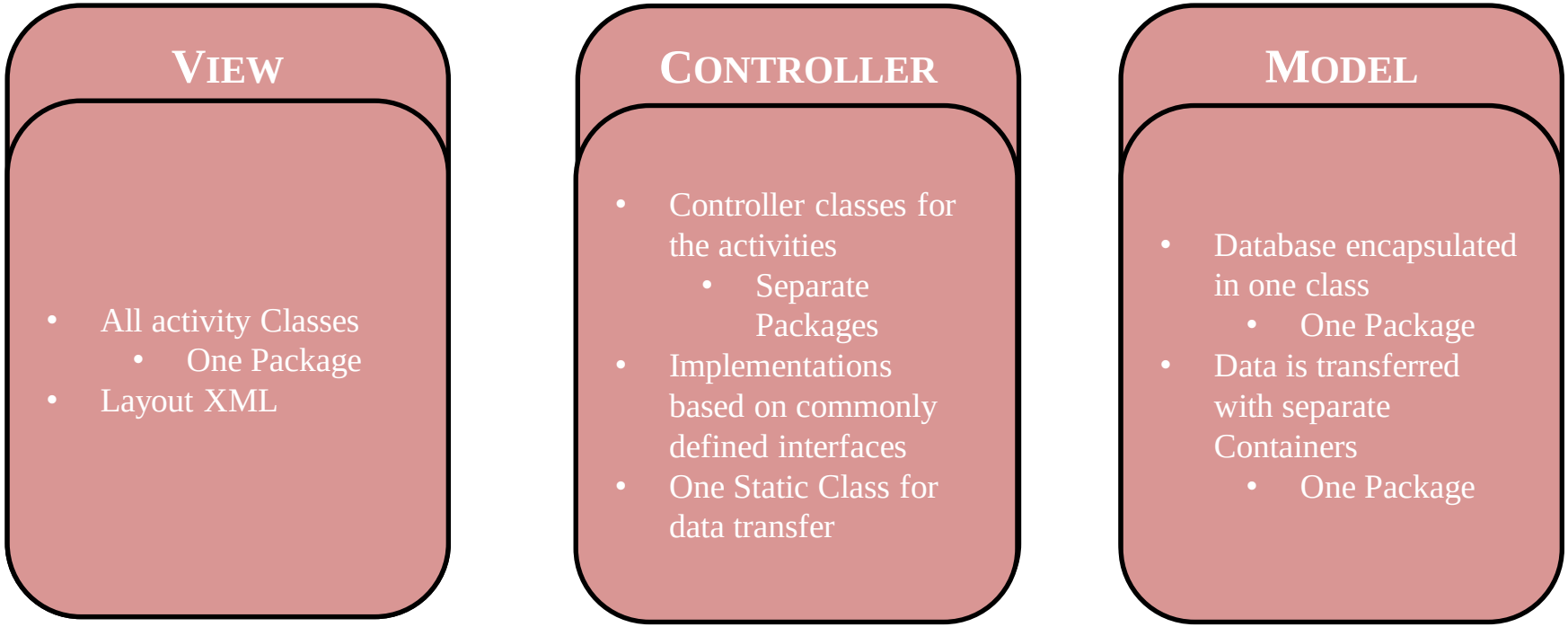
Where are we now?



Conclusions:

- Based on the identified impediments new tasks where defined to achieve the sprint goal
- The new technical architecture was established and further clarified
  - Added new Static class for Session and controller handling
- Further focus on implementation

- **Redesigned technical architecture:**



VISION ESTABLISHMENT *15.11.2015 – 27.11.2015*

TIMELINE

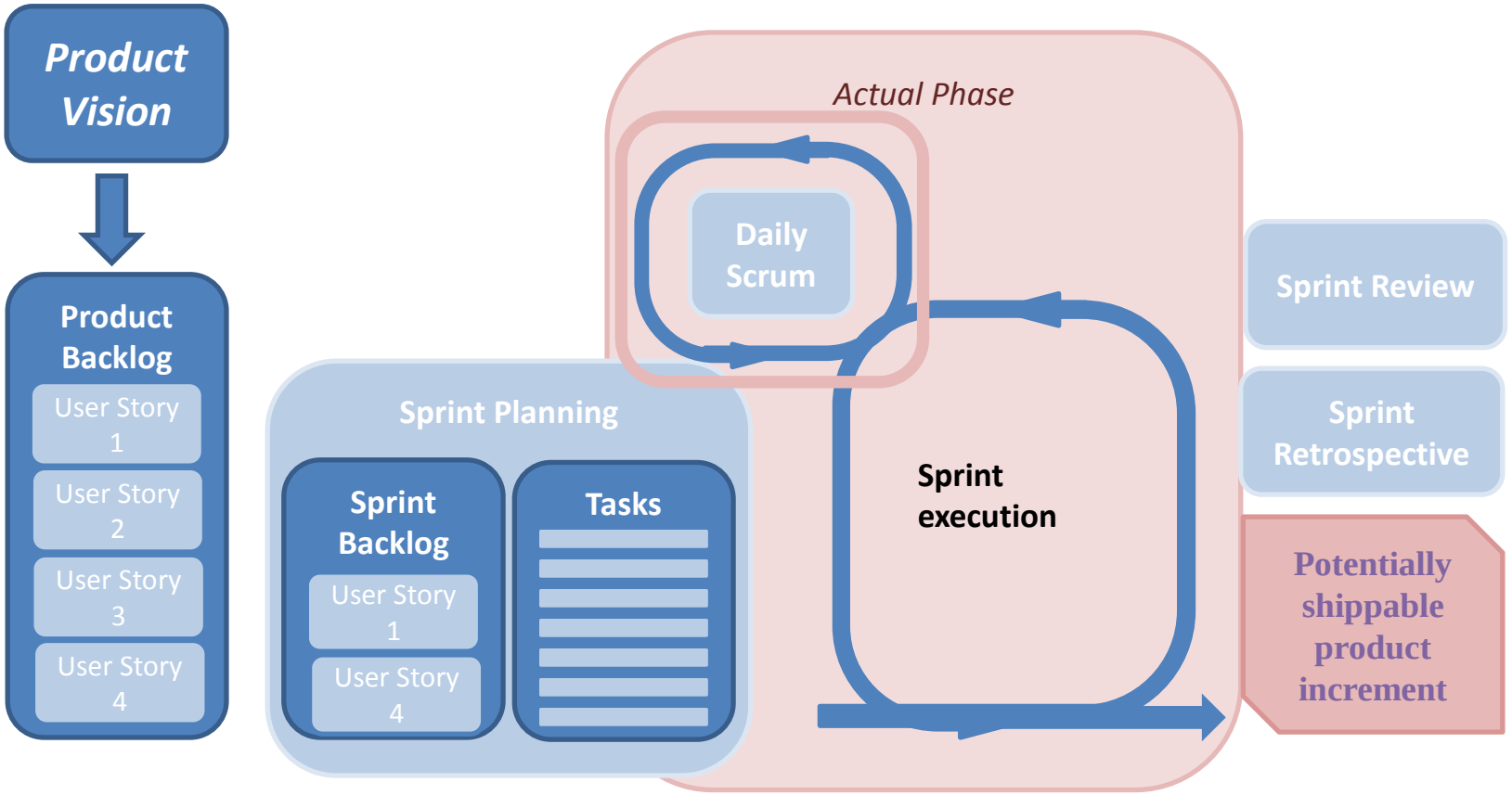
SPRINT I *27.11.2015 – 08.01.2016*

- PLANNING THE FIRST SPRINT *27.11.2015*
- DAILY SCRUM *04.12.2015*
- DAILY SCRUM *11.12.2015*
- **DAILY SCRUM *18.12.2015***

SPRINT II *09.01.2016 – 22.01.2016*

SPRINT III *22.01.2016 – 05.02.2016*

Where are we now?

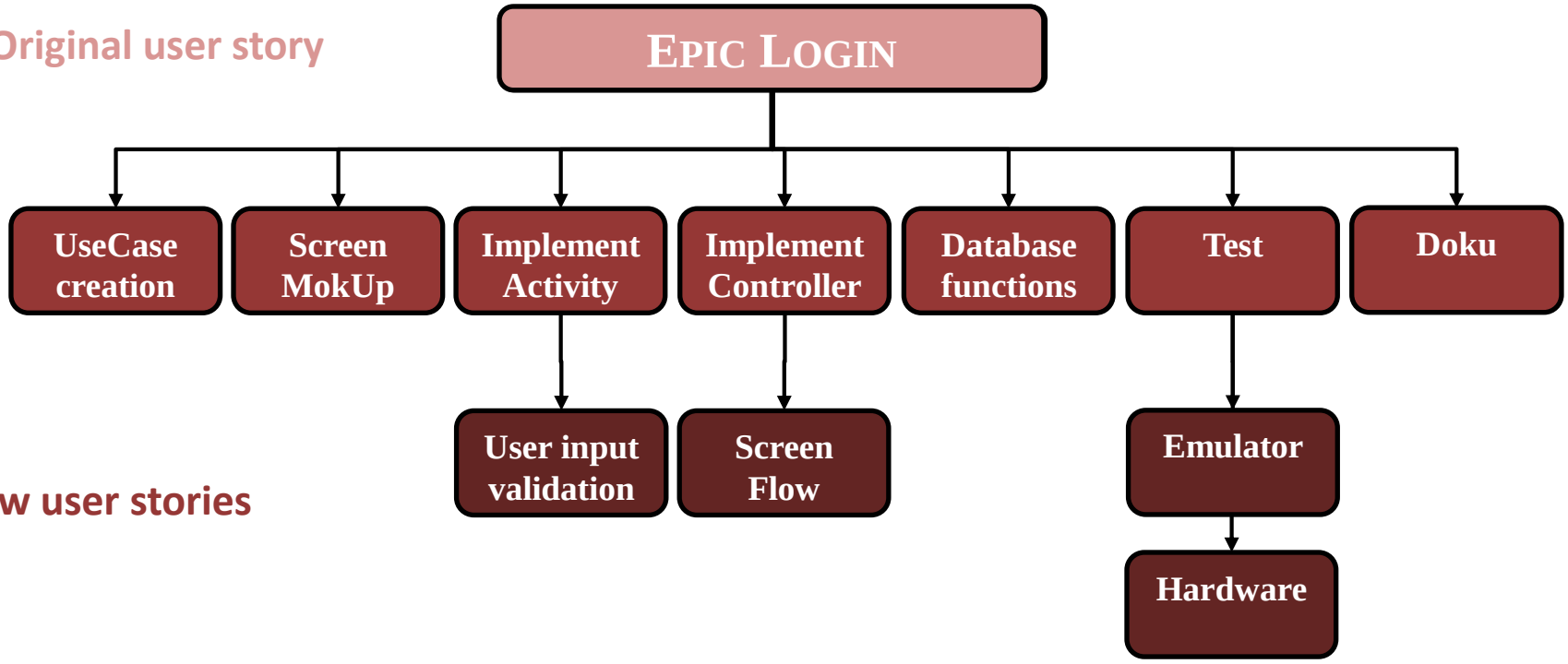


Conclusions:

- Focus on image processing requirements
- Decision to not end the current sprint for the holiday season
  - Additional time will be used on further implementation based on personal preferences
  - Also ongoing rework to reduce scope of user stories

- **Sample of the Transformation of user stories to epics:**

Original user story



New user stories

Tasks defined in sprint planning

VISION ESTABLISHMENT *15.11.2015 – 27.11.2015*

TIMELINE

SPRINT I *27.11.2015 – 08.01.2016*

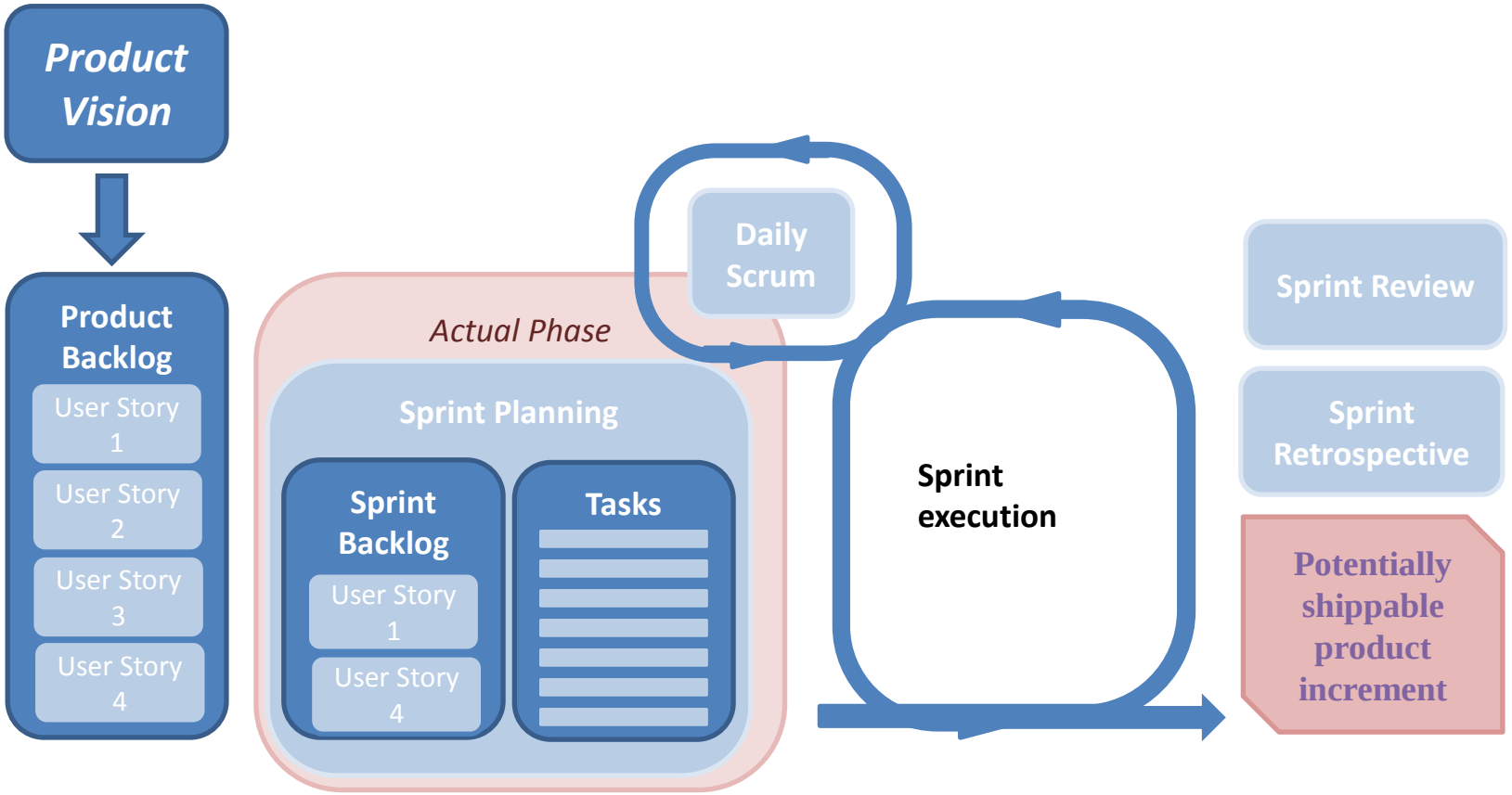
SPRINT II *09.01.2016 – 22.01.2016*

SPRINT PLANNING *09.01.2016*

DAILY SCRUM 15.01.2016

SPRINT III *22.01.2016 – 05.02.2016*

Where are we now?



The Sprint Goal:

- Main goal for this sprint is the integration of the tesseract OCR component
- Furthermore the implementations regarding the basement of the app (i.e. container classes and database model) need to be finalized
- Also additional requirement clarifications and more detailed descriptions are intended

VISION ESTABLISHMENT *15.11.2015 – 27.11.2015*

TIMELINE

SPRINT I *27.11.2015 – 08.01.2016*

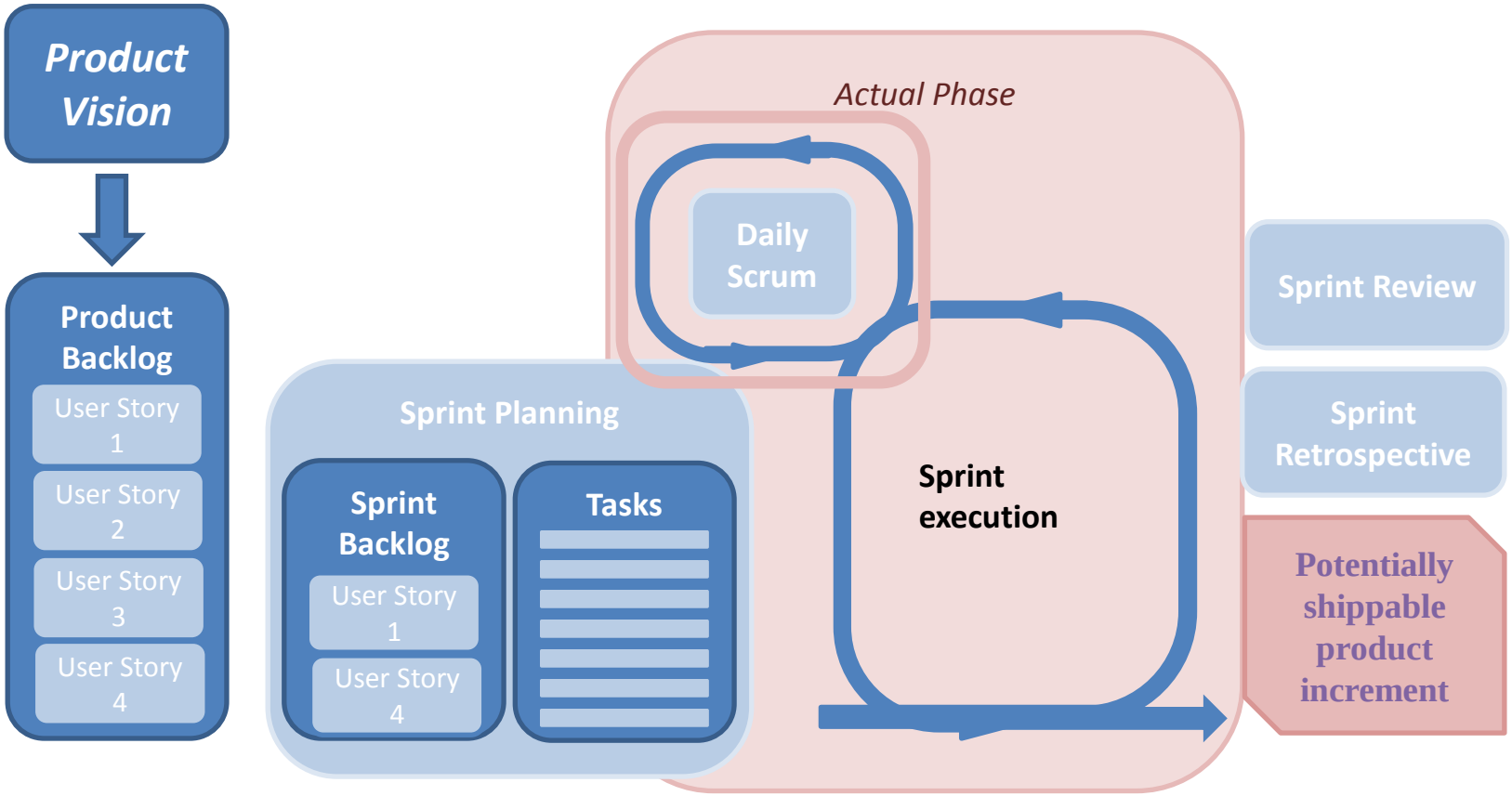
SPRINT II *09.01.2016 – 22.01.2016*

SPRINT PLANNING *09.01.2016*

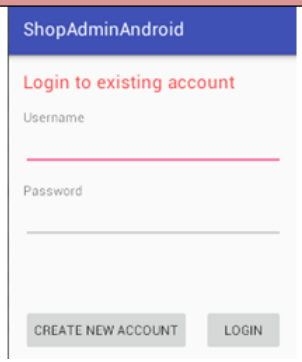
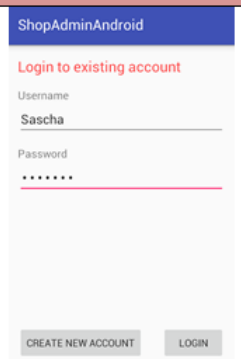
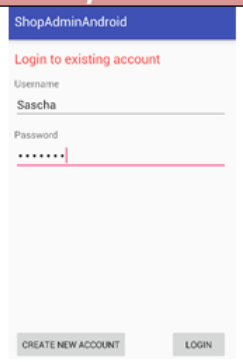
DAILY SCRUM 15.01.2016

SPRINT III *22.01.2016 – 05.02.2016*

Where are we now?



Problem regarding the layouts on various mobile sizes:

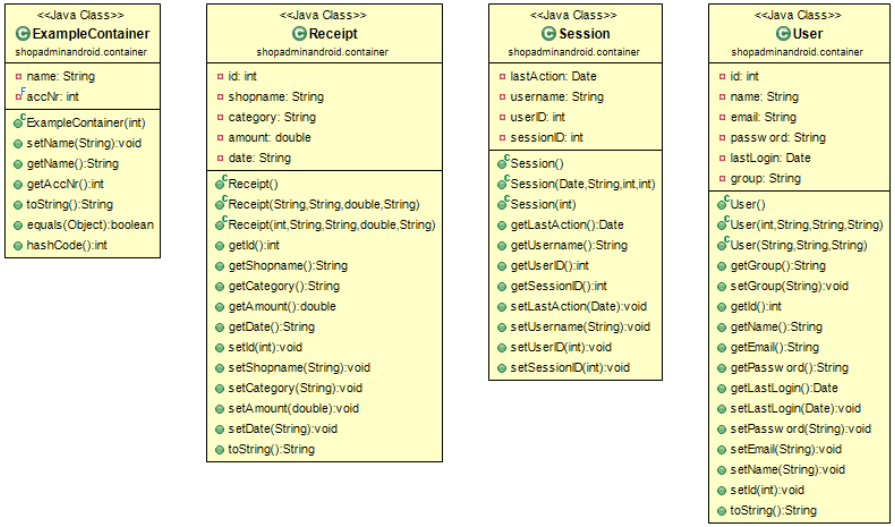
Screen	Small Screen/Low density	Normal Screen/High density	Large Screen/Medium Density	Extra Large Screen/Extra High Density*
Login				

Recommendation within Android SDK is to create individual layouts per mobile size.

Next steps:

- Another validation with Table Layouts
- Focus on normal screen sizes

UML Diagrams created:



Conclusions:

- Focus on normal screen size as no added value is generated on short hand
- Tesseract OCR successfully included into Git-Repository
    - Functionality needs to be implemented
- Still working on velocity so new user stories will be introduced to the sprint
    - Report Requirements, Corporate Layout

VISION ESTABLISHMENT *15.11.2015 – 27.11.2015*

TIMELINE

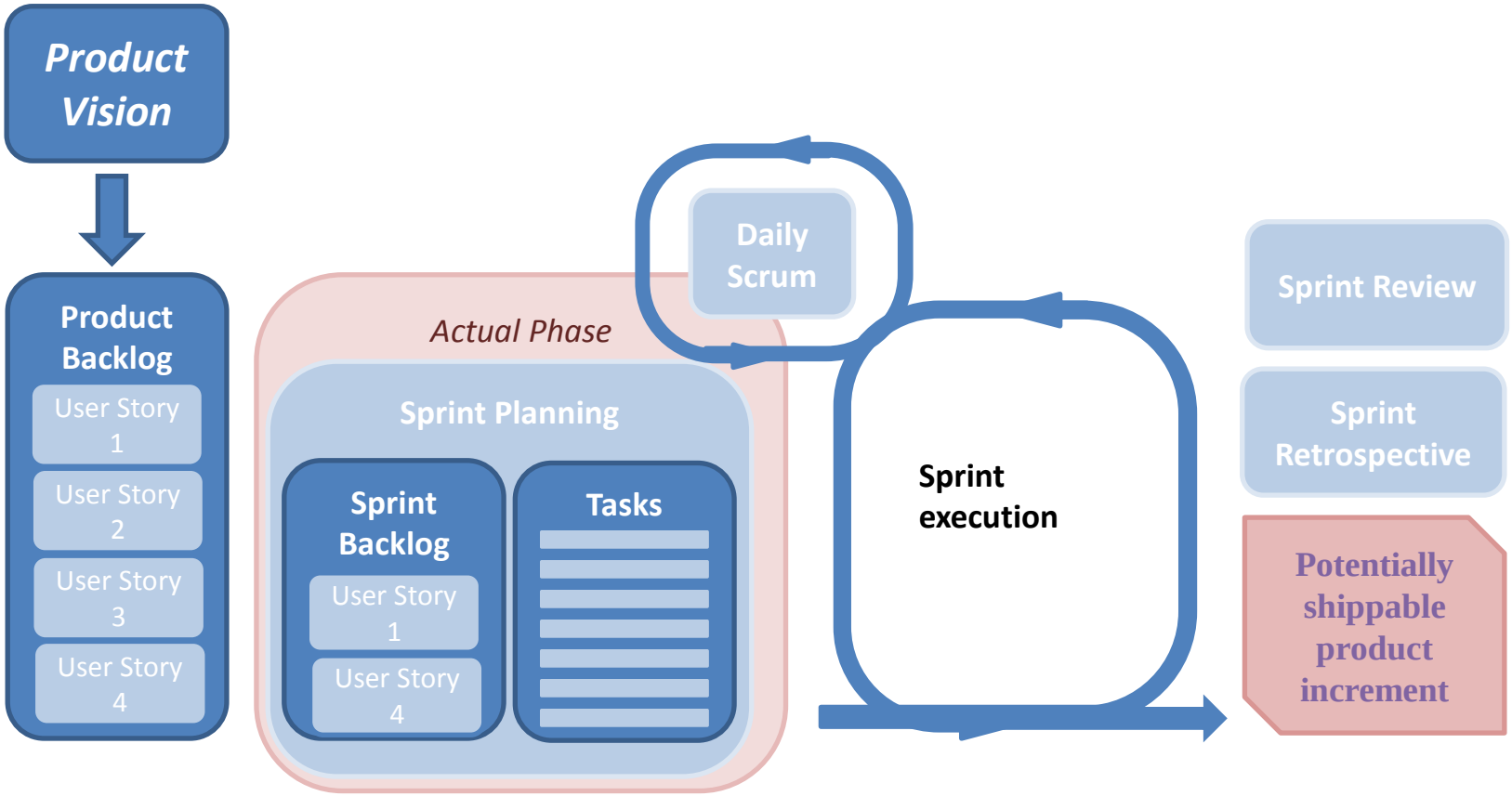
SPRINT I *27.11.2015 – 08.01.2016*

SPRINT II *09.01.2016 – 22.01.2016*

SPRINT III *22.01.2016 – 05.02.2016*

SPRINT PLANNING *22.01.2016*

Where are we now?



The Sprint Goal:

- Maingoal is to receive the OCR Result from tesseract and start with the Classification
- Common view of Burger Navigation
- Furthermore smaller implementations regarding reports, sceenflow and automated testing