Report.md

Initial Planning

Our "Scrum Master" was a normal member of the team, except with a few added responsibilities. Firstly and most importantly, our Scrum Master planned meetings, considering the time and location constraints of individuals as well as time left for the project. For example, most of our team members had directly conflicting schedules, so our Scrum Master instead insisted that the group meet once a week electronically via Skype early in the morning for planning meetings. Our Scrum Master also frequently consulted individual team members outside meetings, taking and sharing notes about any important status updates (i.e. completion) as status updates were infrequent. This and the meetings meant that the Scrum Master was a note taker during meetings.These notes would be used during status meetings and updated as needed. Details were added since status meetings were more detailed and included progress and not just milestones. Tasks were sized based entirely on difficulty. A small sized task could be completed in an hour, medium sized task in less than a day and large tasks spanned over more than one day. During our initial planning meeting, our metrics for an easy task and difficult tasks were based entirely on estimates based on our abilities, as well as what each task encompassed. Our entire team, including the scrum master agreed on the size of the task by consensus, and little argument was raised over the size of the task.

Sprint Backlog

Our project is an Android chatting application that uses Bluetooth instead of the traditional WiFi or cellular data.

For this sprint we planned to accomplish the following for a basic P2P chat that displays a complete connection between the UI and the back-end Bluetooth code (backbone of our app):

1. Basic UI(Opening screen, options to search for Bluetooth devices etc.)
2. Scan for nearby Bluetooth devices
3. Pair with a particular Bluetooth device from the device list.
4. Select that device so a chat bar can be opened.

Our team divided this sprint into four major parts:

- UI - https://github.com/csc301-fall-2015/project-team3-L0101/issues/44
- Scanning for nearby devices- https://github.com/csc301-fall-2015/project-team3-L0101/issues/42
- Pairing to a desired device- https://github.com/csc301-fall-2015/project-team3-L0101/issues/47
- Sending/receiving messages to/from that device- https://github.com/csc301-fall-2015/project-team3-L0101/issues/45

We divided our whole group in half to start working on the front-end and the back-end.

- The front-end consisted of: creating activities/layouts, navigation between activities, sending/receiving chat messages via text file (front-end)
    - Sneh, Alex, Taras, Siddharth
- The backend consisted of: getting the general Bluetooth connections to work and send traffic between sockets
    - Siddharth, Priyen, Akshay, Taras, Sneh

The order of tasks that were to be completed were:

1. UI was completed, the basic layouts of the app were ready and the navigation between activities were taken care of. Since the Bluetooth backend connections were not yet implemented, we used a text file as the “stand-in” for the basic user interactions (i.e. sending messages).

    The user sending message → write to text file
    The user receiving message → read from text file

    Done by: Alex, Taras, Siddharth, Sneh

2. The other three were done simultaneously with each person contributing to some part of the backend
    - Taras implemented the scanning of devices
    - Taras and Siddharth implemented the pairing of devices
    - Akshay laid out the foundations for the underlying Bluetooth connectivity with little help from Sneh
    - Akshay and Priyen worked on the Bluetooth socket connections (sending/receiving messages between sockets)

Meeting notes

19 Oct. 2015

Initial planning meeting

Goal: Discuss finer details such as important classes and what IDE to use.

Progress:

None code-wise since this is the first meeting. Interpreted artifacts and discussed how to implement.

Notes:

- Front end = Activities (Title, Connect, Chat) <- Sneh, Taras, Alex
- Back end = Classes (Bluetooth Controller, Device, Connection List) <- Akshay, Priyen, Taras
- Split into "cells" of three, one for front end, one for back end; cells may be reallocated depending on progress/requirements

To do (expected ETA: end of week):

- Front end: Create activities, make buttons link to other activities, placeholder message sending

- Back end: Create BlueTooth Controller and Device.
- Priority: P2P connection. The ability to chat P2P. These are the bases of our software.

23 Oct 2015

Status meeting

Goal:

Progress:

- Activities created with nonfunctional buttons.
- Basic interface for Bluetooth Manager (unimplemented connect, send message, manage connections) created.
- Significantly less progress than expected or desired.

Notes:

- All group members had particularly busy week, thus not as much time committed as should have been.
- Tasks pushed back to next week (Due 31 Oct 2015)
- Integration of front and back end required; however front and back end need to be made first

To do (by 31 Oct 2015):

- Front end: Create activities, make buttons link to other activities, placeholder message sending
- Back end: Implement Bluetooth Controller and Device

31 Oct 2015

Status meeting

Goal:

Progress:

- UI updated to "send" messages to partner by printing to a ListView in ChatActivity.
- TitleScreen button functionality added; these open up new activities
- Bluetooth Controller checks device Bluetooth on/off status, can connect.
- Bluetooth connect thread class created

Notes:

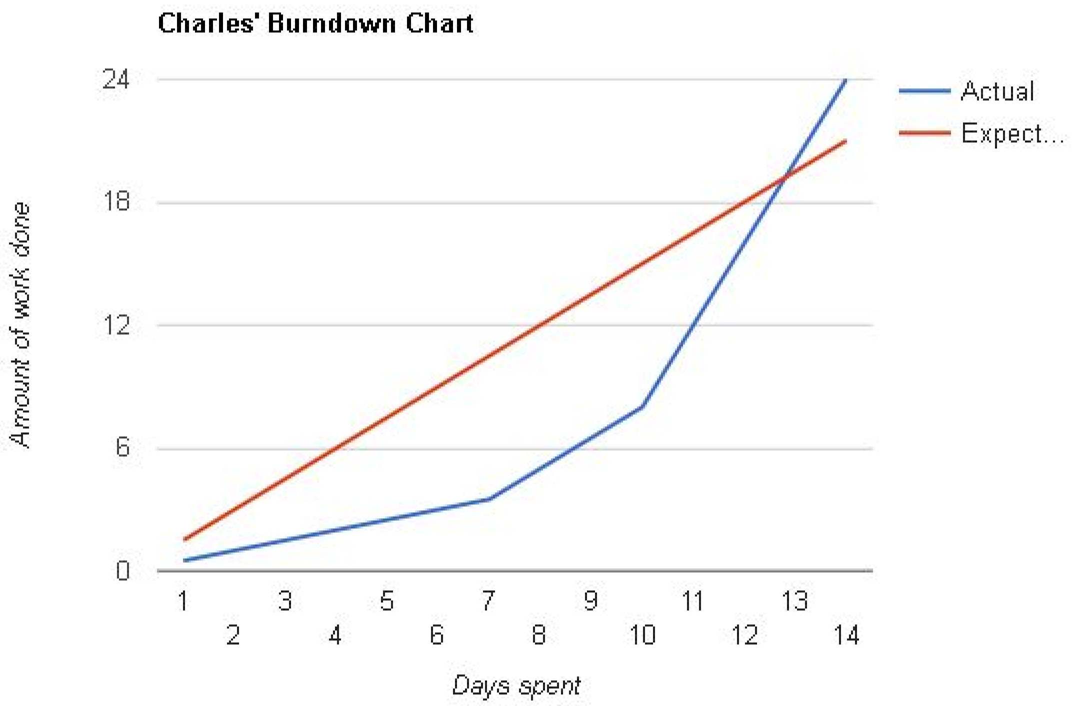
- Front end and back end need to be connected
- Serialization of objects explored as a possibility. Specifically passing client object to ChatActivity.
- Message class needs to be created to store Device object sender as well as the sent message.
- Message class will NOT use a username; device UUID used instead

To do (by 2 Nov 2015):

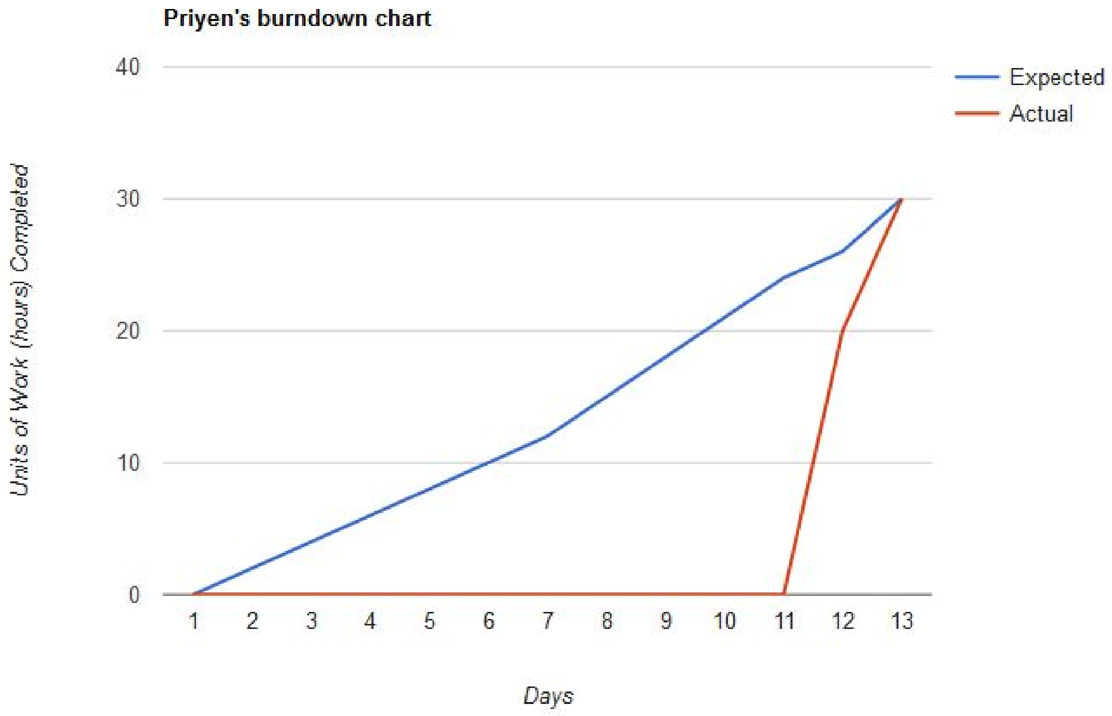
- Integrate front end and back end
- Get P2P chat working (our basic product for the demo!)

- If time permits, do group chat

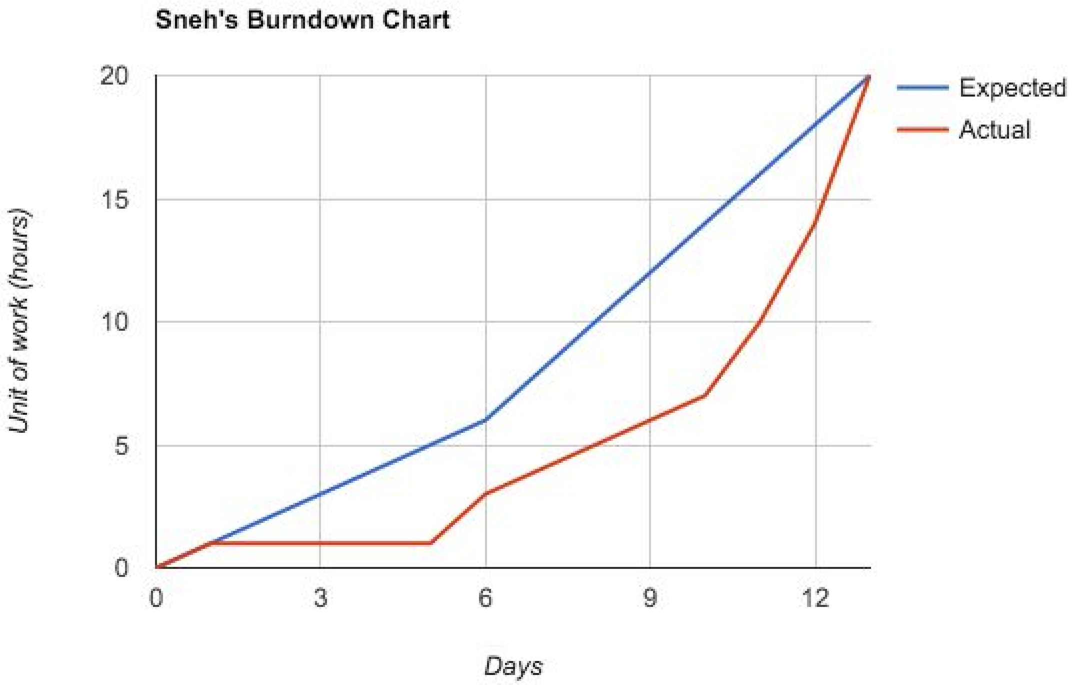
Burndown Chart:

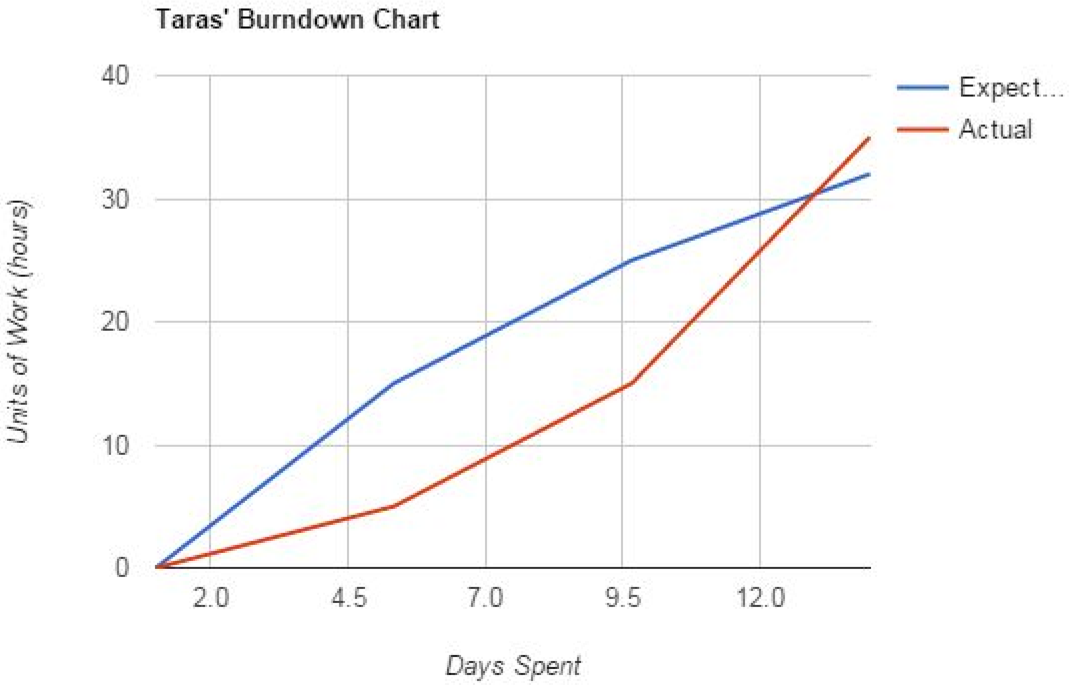


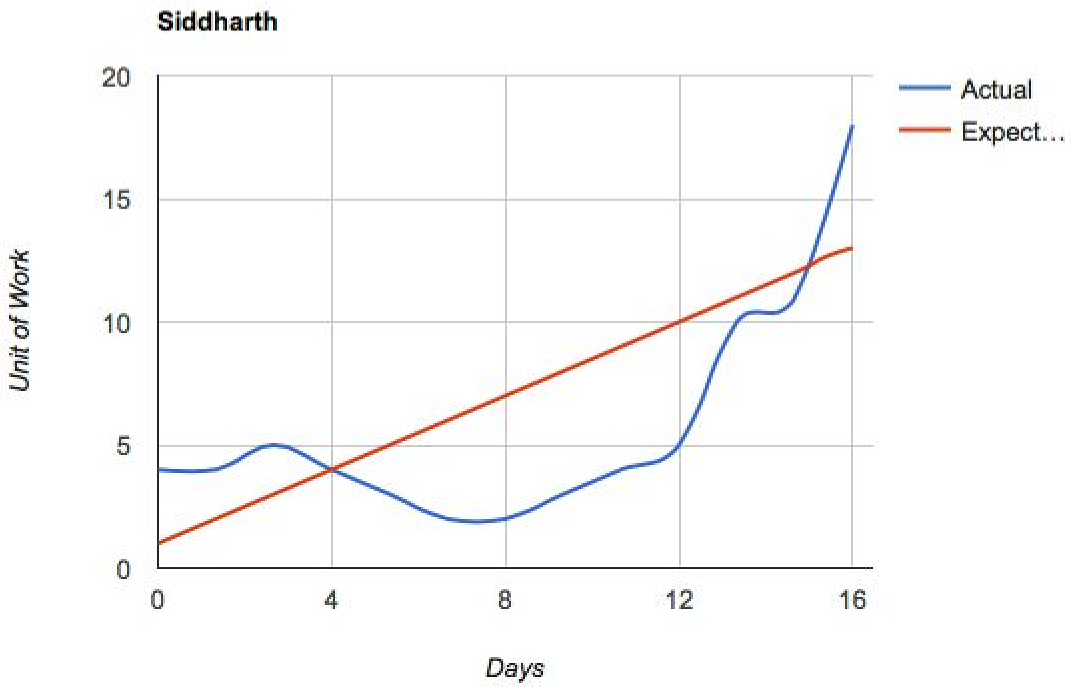
Charles' Burndown Chart

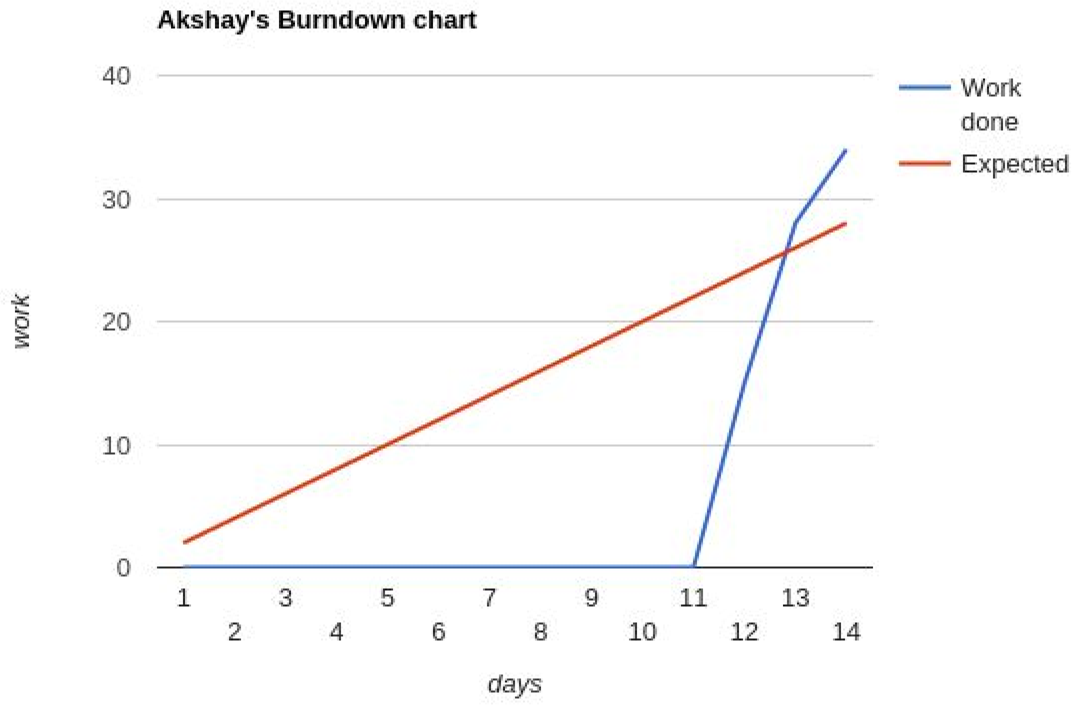


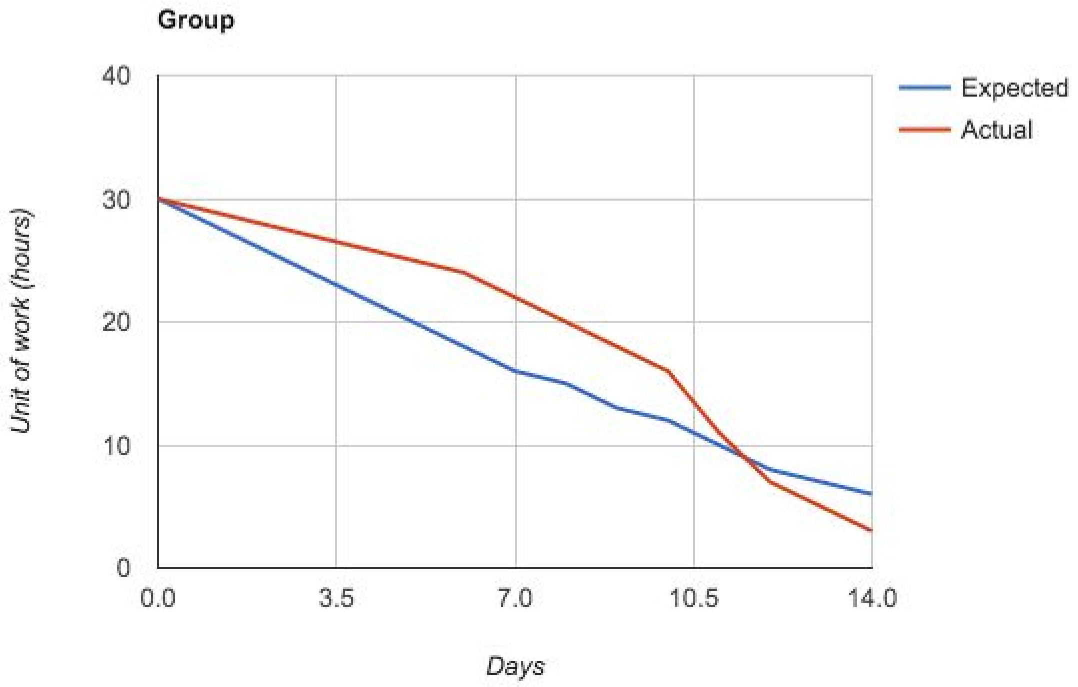
Priyen's burndown chart


Sneh's Burndown Chart
20
15
10
5
0
0
3
6
9
12
Days
Unit of work (hours)
Expected
Actual


Taras' Burndown Chart
40
30
20
10
0
2.0
4.5
7.0
9.5
12.0
Days Spent
Units of Work (hours)
Expect...
Actual


Siddharth
20
15
10
5
0
0
4
8
12
16
Days
Unit of Work
Actual
Expect…


Akshay's Burndown chart
40
30
20
10
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
days
work
Work done
Expected


Group
Expected
Actual
40
30
20
10
0
0.0
3.5
7.0
10.5
14.0
Unit of work (hours)
Days

Review and Retrospective

The first week was extremely hectic for the group members. Many team members had numerous midterms and work-intensive assignments due that week. This diverted attention from the project. However, this meant that not enough work was done on the first week. Therefore, we had to move all the work to the second week. This led to the front-end and back-end classes not being completed the first week and thus severely grew the backlog for week 2.
On the second week, some members had midterms and assignments due. This was far less a factor, however, and the front end was mostly completed mid-week. However, the back end was a much larger task than expected, and required the front-end cell to divert and work on the back end instead, so that the back-end front-end integration could be done. The back-end task of making a Bluetooth Controller class had to be broken down by each of its methods. Furthermore, a decision was made not to implement the username for the message class, because we found that for a basic functional product, this did not need to be added. However, this will most likely be added in phase 3.

Things we could have improved:

1. The first week had very little done due to a lack of focus. It would have helped if every member could contribute more to the group work particularly in this period. This would reduce workload on the second week and make it easier to implement more features as less long periods of coding would be required. In particular, the scrum master should have prompted members to work on the assignment, though the other members should taken initiative and begun earlier.
2. The level of detail in our meetings could be improved. In our initial meeting, we discussed front end and back end, but no planning tools were used (i.e. CRC cards), so it was up to the front end and back end cells to decide what methods to implement, what they needed to do, and how to design them such that integration is as simple as calling a few methods. We also didn't assign individual tasks, leading to chat program based lower-level coordination.
3. We should have dedicated more time to the back end development. Since we agreed that the back end is the more code-intensive section, we ended up diverting a member mid-week to work on the back end instead of the front end. This caused organizational problems, as the tasks were split up for three people for front end and three people for back end.

Things we should repeat:

1. Electronic meetings: Though they lack the literal on-the-spot value of a face to face meeting where the person speaking must stand up, a similar effect was achieved by having everyone shut off their microphones with the exception of the person speaking. Though we could not draw on a sheet of paper to show diagrams/UI for our design ideas, screenshots of UI as well as pictures of drawings sent over the internet were just as effective at showing pictures. The ability to actually communicate live despite the fact that we were physically separated meant that meetings were capable of including all

members, and we felt that the ability to meet at a time and location more convenient for all members far outweighed the on-the-spot atmosphere created by a face-to-face meeting.

2. Assigning optional tasks in a sprint: As the time required by tasks is extremely unpredictable, this gives those who finish unexpectedly early an important feature to implement. This reduces the future backlog as well as ensure that people are not left with no tasks to complete because of an incorrect prediction about the time required to complete a certain goal in a sprint.