



Lesson Assignments in Modern Software Development IV1303

Emil Ståhl
Vårterminen 2019
KTH

Assignment 1

Customer: Infrastahl AB
Team leader: Emil Ståhl
Developers: Emil Ståhl

Requirements

- The system shall keep track of how many minutes a student spend on studying each day and week.
- The system shall issue a warning when the work is either too low or too high.

Opportunity

Circumstances: Most students either study too much or too little.

Need: Students need to keep track of their hours spent on school in order to make good decisions.

Business opportunity: Create a software system that keeps track of hours spent studying and gives the user a warning when the amount drops below or above a certain level.

Requirements:

Format 1:

Description of requirement	Keep track of studying hours
Rationale	Avoid unhealthy work habits
Reference	The user should be able to keep track of the amount of hour of hours the user spends studying during a day. The user should be able to start/stop a timer when studying starts or stops.
Source	The consumer.

Description of requirement	Overworking
Rationale	To achieve a good balance between studies and free time
Reference	The user need to be notified when they have studied a number of hours to not disrupt their balance
Source	The consumer

Format 2:

As a user I want to keep my work balance between two specific limits so I do not fall into an unhealthy situation.

As a user I want to know how many hours I spend on studying on a daily as well as weekly basis.

Format 3:

Use case ID	001	Pre-condition	The user spends the day studying
Use case title	System keeps track of users studying hours	Post-condition	The user will be aware of the amount of hours spent studying
Actors	User, System	Assumption	The user will actively report the hours spent studying to the system
Flow of events	1, User is studying and report this to the system 2, The system acknowledges this and starts a timer that is running until user stop it.	Priority	10
Alternative events	-		

Use case ID	002	Pre-condition	The user has spent the day studying
Use case title	System keeps track of when user should stop studying	Post-condition	The user will stop studying and get a good work life balance
Actors	User, system	Assumption	The user have configured their limits in the system
Flow of events	1, User is studying and registers every start/stop in the system 2, At a specific time point the	Priority	8

	system tells the user to stop studying due to risk of overworking		
Alternative events	The system miscalculates the time spent studying		

Format 4:

Requirement 1

General Requirement Description

Requirement ID: 001.

Requirement Title: Hours spent studying

Requirement Description: Warn the user that the time limit of x hours is reached.

Requirement Type: New.

Internal/External Requirement: External.

Rational: The user shall not work overtime.

Event/Use case ID: **001:1** Every time the user starts studying the user starts a timer that keeps track of the time, **001:2** reset the timer every night and save the current state in memory, **001:3** the user specifies a work limit in settings.

Related to Requirement(s): None.

Non-functional Requirement: Deadline 2019-04-05.

Constraints: No relevant constraints.

Conflicting Requirements: None.

Intended Users: Students, workers in general, workaholics in particular.

Specific user who stated the Requirement: Infrastahl AB.

Customer Satisfaction: High.

Customer Dissatisfaction: Possibly with that they have to start and stop timer every time.

Reference documents: None.

Requirements Evaluation Data

Business Value: For this application I will not need any cooperation with other companies because I am my own developer. Other value: This application will focus on students, but also other grownups that would like to keep track of their work habits and/or end their addiction to work.

Requirements Priority (Rank): The most important parts that need to be implemented at first is: timer and a button so the user will be able to keep track of its balance. If the developers have the resources the developers could keep track of more types of activities.

Fit Criteria: It should exist some kind of timer and a memory that can keep track of all times.

Risk: There is one mainly risk that will impact the success of this software project; that people find it annoying and disturbing to have to manage another piece of software along with their other work. If all users find this annoying and disturbing, it will be no use of this application.

Other Description Data

System data: The requirement will not have any subsystems or components and will be implemented as a single component of the study app.

Interfacing System ID: The requirement will cooperate with requirement 002

Environment: The requirement will be implemented into the software that is the study app.

Requirements Reporting Data Requirements

Reporting Date: 2019-03-21

Originated by: Infrastahl (Customer)

Reported by: Infrastahl (Customer)

Requirements owner: Developer team

Requirements Reporting Data

Requirements Reporting Date: 2019-03-20

Originated by: Infrastahl AB (Customer)

Reported by: Infrastahl AB (Customer)

Requirements owner: Developer team

Requirements Management Data

Preliminary Implementation Plan: Develop the functionality of tracking studying hours by implementing a timer in an application on your phone where the user reports when user starts/stops studying.

Preliminary Outline of activities: Design the study timer, implement it, test if the app properly keeps track of time spent studying.

Design the functionality:

- *Activity Description:* Design the look and implementation of the study timer by creating a simple wireframe and writing down all necessary coding concepts used to implement the functionality
- *Activity Start Date:*
- *Activity End Date:*
- *Expected/Actual Result:* An idea of how the function will look and how it should be implemented
- *Activity Conducted by:* Developers
- *Activity Approved by:* Developers
- *Effort Spent on Activity:* Roughly 1 work hour
- *Cost of Activity:* 0 costs

Implement the functionality:

- *Activity Description:* Write the code for the study timer.
- *Activity Start Date:*
- *Activity End Date:*
- *Expected/Actual Result:* A functionality in the app that keeps track of studying time.
- *Activity Conducted by:* Developers
- *Activity Approved by:* Developers
- *Effort Spent on Activity:* Roughly 2 work hours
- *Cost of Activity:* 0 costs

Test the functionality:

- *Activity Description:* Conduct tests to confirm correct functionality. Try to press the button and see that the timer starts
- *Activity Start Date:*

- *Activity End Date:*
- *Expected/Actual Result:* A confirmation that the functionality in the app that keep track of time works.
- *Activity Conducted by:* Developers
- *Activity Approved by:* Infrastahl AB (Customers)
- *Effort Spent on Activity:* Roughly 0.5 work hours
- *Cost of Activity:* 0 costs

Requirements Completion Data

Actual completion date: 2019-04-06

Planned completion date: 2019-04-05

Relation to tests: Test 1: Working timer. Test 2: saves the start time when button is pressed. Test 3: The memory does not drop the time spent studying until the reset is issued. Test 4: The reset works at the right time.

Released in:

Requirement completion:

Sign off date:

Signed off by: Infrastahl AB (customer)

Estimated effort: 7 work hours

Actual effort:

Estimated cost: -

Actual cost: -

Post Implementation Data

-

Requirement 2

General Requirement Description

Requirement ID: 002.

Requirement Title: End studying time

Requirement Description: Warn the user that if they study after this time they might get overworked.

Requirement Type: New.

Internal/External Requirement: External.

Rational: The user should not be overworked due to a too much time spent studying

Event/Use case ID: 002:1 A notification is shown at the time specified in the settings by the user. 002:2 If the user changes the time in the settings the notification should be shown at that time instead.

Related to Requirement(s): None.

Non-functional Requirement: Deadline 2019-04-05.

Constraints: No relevant constraints.

Conflicting Requirements: None.

Intended Users: Students, workers in general, workaholics in particular.

Specific user who stated the Requirement: Infrastahl AB.

Customer Satisfaction: High.

Customer Dissatisfaction: The need to specify start and stop time.

Reference documents: None.

Requirements Evaluation Data

Business Value: For this application I will not need any cooperation with other companies because I am my own developer.

Other value: This application will focus on students, but also other grownups that would like to keep track of their working habits.

Requirements Priority (Rank): The most important parts that need to be implemented at first is: timer and a button so the user will be able to keep track of studying time.

Fit Criteria: It should exist some kind of timer and a archive that can keep track of time spent studying.

Risk: It exists one mainly risk that will impact the success of this software project; that nobody would like to limit their studying time. If there is nobody that would like, to limit their working habits it will be no use of this application.

Other Description Data

System data: The requirement will not have any subsystems or components and will be implemented as a single component of the studying app.

Interfacing System ID: The requirement will cooperate with requirement 001

Environment: The requirement will be implemented into the software that is the studying app.

Requirements Reporting Data

Requirements Reporting Date: 2019-03-20

Originated by: Infrastahl AB (Customer)

Reported by: Infrastahl AB (Customer)

Requirements owner: Developer team

Requirements Management Data

Preliminary Implementation Plan: Develop the functionality of warning the user when the time has reached the time specified in the settings.

Preliminary Outline of activities: Design the risk of overwork warning, Implement it, test if the app properly keeps track of when user should stop.

Design the functionality:

- *Activity Description:* Design the look and implementation of the studying habits consumption after this time warning by creating a simple wireframe and writing down all necessary coding concepts used to implement the functionality
- *Activity Start Date:*
- *Activity End Date:*
- *Expected/Actual Result:* An idea of how the function will look and how it should be implemented
- *Activity Conducted by:* Developers
- *Activity Approved by:* Developers
- *Effort Spent on Activity:* Roughly 1 work hour
- *Cost of Activity:* 0 costs

Implement the functionality:

- *Activity Description:* Write the code for the time based warning.

- *Activity Start Date:*
- *Activity End Date:*
- *Expected/Actual Result:* A functionality in the app that notifies the user that they should not study more after a preset number of hours.
- *Activity Conducted by:* Developers
- *Activity Approved by:* Developers
- *Effort Spent on Activity:* Roughly 2 work hours
- *Cost of Activity:* 0 costs

Test the functionality:

- *Activity Description:* Conduct tests to confirm correct functionality. See that the notification comes at the right time. And that it works when you change the set hours.
- *Activity Start Date:*
- *Activity End Date:*
- *Expected/Actual Result:* A confirmation that the functionality in the app that warns overworking.
- *Activity Conducted by:* Developers
- *Activity Approved by:* Infrastahl AB (Customer)
- *Effort Spent on Activity:* Roughly 0.5 work hours
- *Cost of Activity:* 0 costs

Requirements Completion Data

Actual completion date: 2019-04-06

Planned completion date: 2019-04-05

Relation to tests: Test 1: Notification works at the right time. Test 2: Change time and check that test 1 still works. Test 3: The memory does not drop the set time.

Released in:

Requirement completion:

Sign off date:

Signed off by: Infrastahl AB (customer)

Estimated effort: 7 work hours

Actual effort:

Estimated cost: -

Actual cost: -

Post Implementation Data

-

Comparisons of formats:

Format 1-3 all only include the barebone information of the requirement. It describes in a short and concrete way what the requirement is about and why it exists. Out of these format 3 also provides an idea of the implementation and how the interaction between the user and the system will happen. Format 4 goes into detail of the requirements, answering questions like who implements, tests or approves the requirement, different important dates and things like constraints and risks.

A natural consequence of the very detailed format 4 is that it take a lot more effort and is also more difficult to write than the other 3 formats. These formats doesn't take a lot of effort or is very difficult to write.

Requirements

Conceived: The alpha is fulfilled. I have been able to check almost all the points at this alpha, that is specifying the user, the stakeholders and the opportunity. Since there is no clear funding stakeholder this point is yet to be crossed

Opportunity alpha

Identified: The alpha is fulfilled. The stakeholder is identified.

Requirement item

Identified: The item is fulfilled. Since I have documented the stated requirements well all the points are covered and the origin of the requirement item is clear.

The “Funding stakeholders identified” checkbox might be hard to understand since it might not be clear who this is. At the start of the whole process there might be people interested in funding the project but not yet decided before more research and project prototyping has been done.

Assignment 2

- **Test Case ID 001.**
- **List of Requirements:** When the user press the study button the timer starts counting.
- **Input:** The user is pressing the study button.
- **Output:** The timer is running.
- **Environmental needs:** None, other than it requires someone to press the button.
- **Inter-case dependencies.** None.
- **Special procedural need.** None.

- **Test Case ID 002.**
- **List of Requirements:** ID 001
- **Input:** User starts studying and user presses study button
- **Output:** Study timer is activated and time is displayed on the screen as output.
- **Environmental needs:** Requires that a user presses the study button
- **Inter-case dependencies:** Case ID 001
- **Special procedural need:** -

- **Test Case ID 003.**
- **List of Requirements:** -
- **Input:** The setting button is pressed.
- **Output:** The settings page is reached.
- **Environmental needs:** -
- **Inter-case dependencies:** -
- **Special procedural need:** -

- **Taste Case ID 004.**
- **List of Requirements:** ID 002
- **Input:** The time reaches the hours specified in the settings.
- **Output:** The textbox-component outputs a warning.
- **Environmental needs:** That the amount of hours has reached a preset number.
- **Inter-case dependencies:** -
- **Special procedural need:** -

- **Taste Case ID 005.**
- **List of Requirements:** ID 002
- **Input:** The time reaches what time you have set it to reset
- **Output:** A reseted study time.
- **Environmental needs:** That the time has reached a certain time
- **Inter-case dependencies:** Test Case ID 003
- **Special procedural need:** -

Step 1.

Requirement 001: It will probably take a little bit longer to implement than specified earlier. I had guesses between ½ hour to 8 hours. Probably I will have to plan that this part will take 4-8 hours to have done.

Requirement 002: This requirements is probably going to take more time than the requirement above, I had guessed between 8 hours and 13 hours.

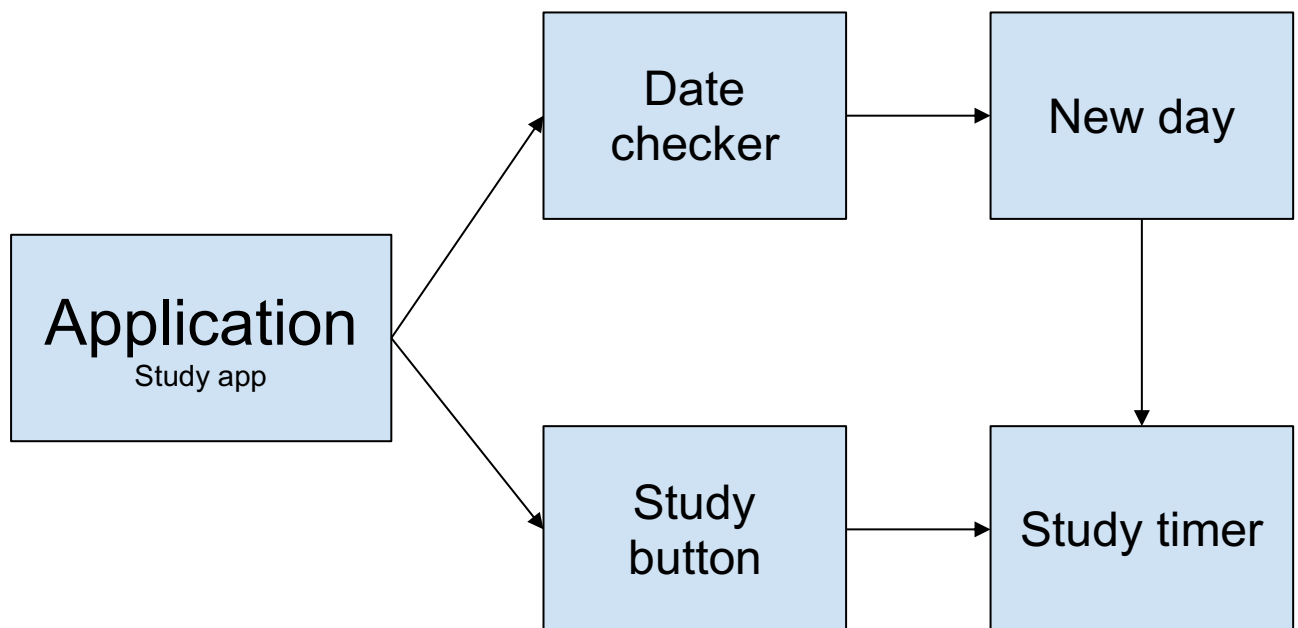
Step 2.

The requirements can be divided into three low-level requirements:

- That by pressing the study button the timer will start. This requirement is estimated to be around 100 LOC in size and take approximately 2 working hours.
- That when reaching a certain number of hours the system will react in certain ways. This requirement is estimated to be around 50 LOC in size and take approximately 2 working hours.
- That when the time has reached the values predefined by the user the system will tell the user to stop studying. This requirement is estimated to be around 50 LOC in size and take approximately 2 working hours.

Traditional Approach vs. Agile Approach

The traditional approach of estimating size and effort is hard and gives a fairly uncertain result since it is very hard to estimate something before you have done it. It is also hard to give a good answer of exact lines of codes, it is more or less impossible to predict beforehand. The agile approach on the other hand doesn't require that you give an exact answer but rather makes way for good and necessary conversation. You will most likely not have the same answers which means that people with different views on the problem can explain their perspectives on the matter. This will result in a discussion about estimated size and effort that covers a broader area of the requirement at hand, which will give better overall results.



Software system alpha

All points of the first alphacard are crossed. Since the software system is simple and doesn't include a lot of different parts things like system boundaries and system organization is easy to identify. I will be using a data-flow architecture and the hardware used will be a standard smartphone. I'm not buying or reusing anything and there are no major risk with the system.

Team alpha

In the team alpha most points are covered. Since I am the only developer I will do both coding, documentation and testing. Since the project is not very extensive things like growth mechanisms and governance rules have not been decided, and will probably not be.

Requirements alpha

In the requirements alpha some more points have been crossed on the second alpha card. The development stakeholders has been identified and the purpose and success of the system is clear. How the requirements will be formatted and changed has also been decided, it will be simply be by changing a Word Document.

Requirements item

In the requirements item all of the points of the “described” phase has been fulfilled when completing Assignment 1. The requirements have been described in detail.

Opportunity alpha

In the opportunity alpha all of the points on the “solution needed” card have been crossed since the solution of a web application have been identified. Some of the points of the “Value Established” has also been crossed as the impact and value of the solution has been understood.

Assignment 3

The main purpose of a scenario is that it provides a great understanding on how to plan a project and tackle different parts of the project. In a project like this it can be extremely hard to get a grasp of all the things that is needed. That's where the cards come in. The cards gives a clear visual understanding of the progress of the project and which areas that needs the most focus at every given moment.

A lot of studies have shown that the most important factor for a project with more than one participant is communication. It's often hard to implement a theoretical methodology in real life exactly as the inventor thought of it. I think it's great if a team is able to implement the same communication in real life but at the same time I think it's important to have these concepts as a starting ground to stand on but at the same time be open for new ideas on how to structure work for the most effective and optimal results.

I think the method is quite straight forward, I do not have any problems with understanding the concepts. I just noticed that the order is not that important of the planning (as in the Scenario). You just have to make sure that you make it through all the check-points, other than that I think you can do the cards in any order.

The perks of Planning Poker

In a group with more than just a few members it can sometimes be hard to have a discussion where every member is equally heard. Some members may take a more intensive role and others may be more laid back. This is were planning poker comes in.

Planning Poker makes every team member think for themselves and then present their opinion to the group, this makes sure that every member is given equal power to present their view of what's currently being discussed. By using this method, there is a higher chance that every member presents their honest view and doesn't get influenced by other (more active) members. It's also easy to directly see how the opinions vary in the group by just simply take a look at the cards.

As with most things, there exists downsides with planning poker. One of them can be that it takes some time to get through. The method is therefore not optimal for large teams. In addition to this, it can take more than one round to get the majority of members to agree.

Impression of grasping the whole domain of software engineering with SEVEN ESSENTIAL THINGS – ALPHAS.

In a project there can easily arise a million things to do. That's were The Seven Essential Things comes in. It's a great way of keeping track of all important things in a project. It's impossible to have it all in your head, and even if it was possible that's not a great way of collaboration. It's necessary that every member has access to all the information and knows what to do.

A clear visual system gives all team members a peace of mind when it comes to planning their work. It will prohibit a lot of misunderstandings and minimize workload due to a clear path of work since you'll always know what's next in the project and what's missing. I think it's nice with the visual representation that you get from the alphas.

The checklist makes sure that every little thing gets noticed since it's often the bigger things that gets the most attention. This can be things like defining risks or clearly define the goals. Things like this might be example of things that can easily be overlooked due to that this kind of teams consists of mainly hard core developers that enjoy coding more than everyting else in this world. A checklist gets these things done.

Assignment 4

Requirement implemented: ID 001, keep track of study time and display to user when limit is reached.

Effort estimated: Approximately 2 hours

Risk analysis: Due to this small requirement it's hard to find any obvious risks. The risks that does exists is mainly about time management, i.e. that the requirement not being finished. But there does not exists any risks that could affect the results.

Requirement tasks:

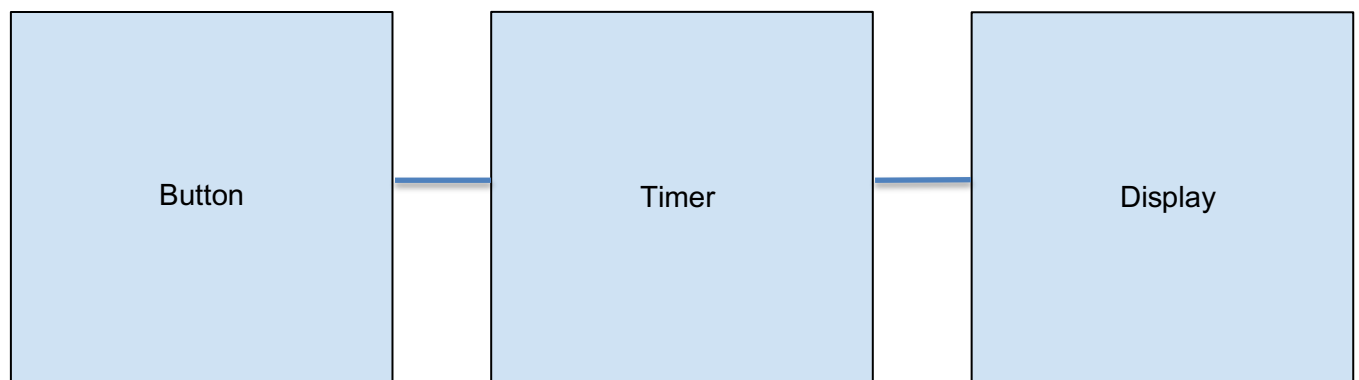
- Task 1: Study timer, goal is that you should be able to click a button and by doing so, start a timer that's running until you stop it or limit is reached.
- Task 2: Display current time, the goal is that the user should be able to see the current value time spent on studying.

Data that needs to be managed: The time spent studying.

Steps to be processed:

1. Implement a button
2. Make button start a timer on click
3. Visualize the study time on the screen.

Graphical Method



Pseudo code

- Initialize study timer to 0
- While (active)
 - if (buttonClick)
 - start timer
 - Display time on display
 - Run until stopped

Test cases: Click on button and make sure that the display is updated properly (timer running).

Requirement implemented: ID 001, keep track of studying time and display to user

Effort estimated: Approximately 8 hours

Risk analysis: The only risk is that the estimated effort is too low and won't be enough to implement this requirement.

Requirement tasks:

- Task 1: Study timer, goal is that you should be able to click a button and by doing so, activate a timer that runs until stopped.
- Task 2: Display study timer, the goal is that the user should be able to see the current time spent on studying.
- Task 3: Store the value of the timer so that user can get a weekly report of time spent studying.
- Task 4: Reset the timer everyday

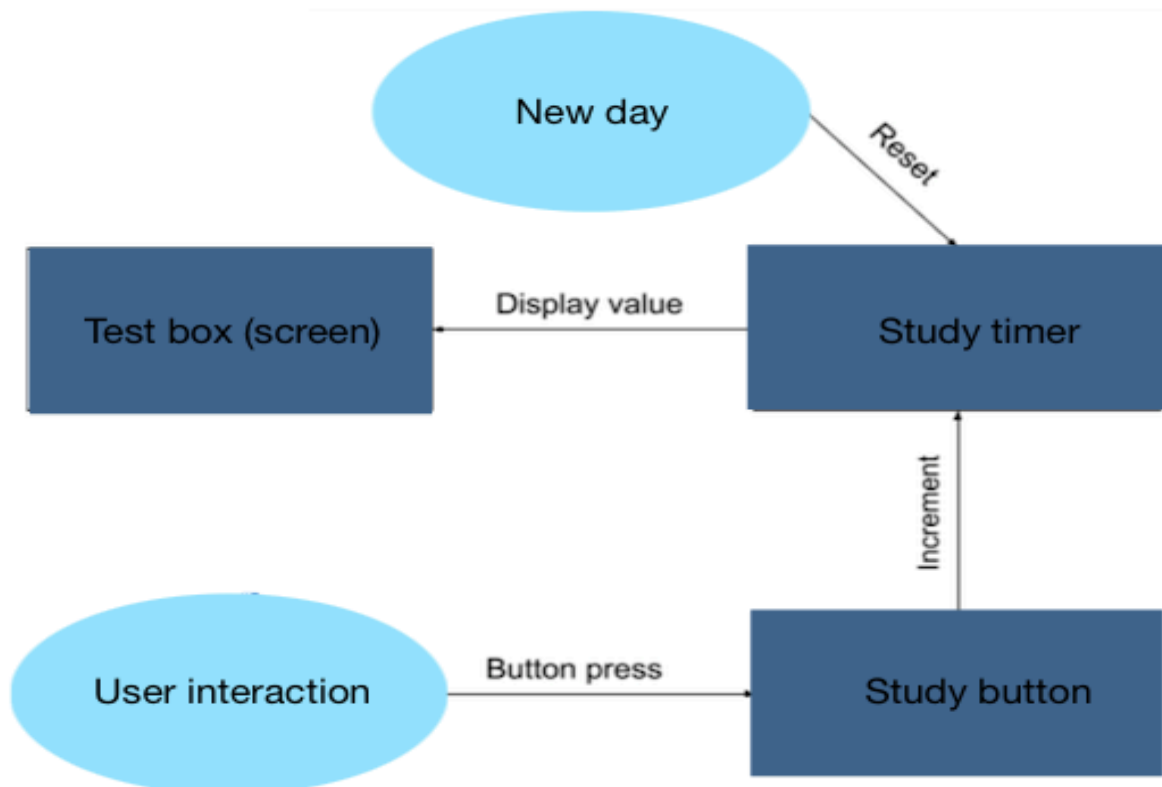
Data that needs to be managed:

- The value of the study timer.
- Each button press
- The current date

Steps to be processed:

1. Implement a button
2. Make button start timer on click
3. Store the counter value locally
4. Visualize the coffee counter in a text box on the screen
5. Reset timer every new day

Graphical Method



Pseudo code

```
function study_button;  
    if user click;  
        start and run timer  
  
function text_box;  
    display study timer which is stored locally  
  
function timeout:  
    if a new day or limit reached:  
        save time to memory annd reset study timer
```

Test cases:

- Click on button and make sure that the timer runs properly.
- Make the timer reset every minute as test

Discussion

Pair programming

For	Against
<ul style="list-style-type: none">• You have the possibility to always ask your co-worker about the code if you don't understand something or how to proceed. You complement each other.• Most projects in the working life includes multiple developers. This is a great way to practice this and learn concepts from each other.	<ul style="list-style-type: none">• Might lead to unstructured code as more than one person works on it. Different individuals has different opinions. Leading to disputes that slows down development.• Can be less efficient to have multiple programmers working on one assignment.

Test-first programming

For	Against
<ul style="list-style-type: none">• Level of progress fully depends on if the work passes the test. Easy to see what works and what doesn't.• Test-driven programming drives the work forward. Aims for solutions to problem.• Easy to find bugs when small features are implemented each time.	<ul style="list-style-type: none">• It can be hard to set up goals before you have started the project and started coding.• People can get too focused on testing and forget to develop new features.

Refactoring

For	Against
<ul style="list-style-type: none">• Improved code readability and reduced complexity; these can improve source-code maintainability• Less code, easier to debug.• Making it possible for people outside project to understand code with the help of comments	<ul style="list-style-type: none">• Refactoring doesn't improve the main features of the software/product. Time intensive that doesn't follow thru to the end customer.• Takes time to comment

Designing before coding

For	Against
<ul style="list-style-type: none">• More efficient coding. If you start coding right away you might make big mistakes that takes time to solve compared to if you design the structure on a piece of paper before start writing code.• Easier to spot mistakes before starting coding. You can focus on making a good structure without worrying about semantics.	<ul style="list-style-type: none">• It can be frustrating to have to do "boring" work before you get to start coding.• It is time consuming.

Estimating effort required

I had estimated that it would take around 2 hours to implement the function but it ended up taking closer to 4 hours due to that I thought it existed some kind of API to implement a timer in java but I had to do a special solution that required more time. A pro tip is to investigate stuff like this before estimating time required in order to get a more precise estimation.

Assignment 5

Emil Ståhl

The big picture regarding this matter is pretty clear. However, if I would go into details I have some thoughts about in which order you should proceed with the Essence Alphas. It's obviously too much work to have to go thru every alpha at every single meeting regarding pain points, but I think it can be a risk of missing important stuff if you proceed with the opportunity and stakeholders alphas to a later stage before you have looked in to the stages of the other alphas?

As a example of this, what would happen if you have not looked in to the alpha of Software System at a late stage in the project? I argue that is quite important to keep an eye on all of the alphas in like a parallell manner in order to not miss something and bring it all together.

If the team have started working on the main product as well as started cooperated with stakeholders without have considered any of the alphas regarding team or way of working. If these alphas have not been fully considered it can bring up serious problems in the team. Things like this lead to a slow down in progress and might result in irritation and lack of sympathy in the team.

Time spent on planning and making strategies often saves a lot of time during the actual work. Therefore I think it could prohibit some problems if the team considered all of the alphas in a parallell way. There is no need to work thru them all at every meeting but I think you maybe could have some meetings that are dedicated to all alphas.

The first scenario was great in order to get a deeper understanding of how to work with the alphas.

Conclusion

The main points of the scenario is quite straight forward to understand but as I described here above there is some reflection about in which order the alphas should be considered and what's results in a alpha can be ignored or not. By discussing with other teams I have got the understanding of that this in some manner depends on the nature of the alpha and it might differ from scenario to scenario. There is also never suffiecent to completely ignore an alpha if you not have a good reason for it.

Scenario 1 – Requirements alpha

- *Concieved state*: Fulfilled – The stakeholder providing funding has been identified and KTH has agreed that the system in question is going to be produced. The users corresponds to the administrators, faculty has also been identified as well. Regarding the opportunity to renew how KTH handles administration is clear as well. On the grounds of this, the concieved state has been fully fulfilled.
- *Bounded state*: Fulfilled – Due to that the university KTH has stated that the system should deal with the administrative work of courses, information and grades means that both the purpose and success of the system is agreed and clear which in order

means that it fully meets the requirement for to be fulfilled. The development team has in this case been identified to be consisting of one person. Requirement format and management is somewhat in place as the team is having continuous meetings with stakeholder and users to make sure that the project proceeds as it should. The identified constraints is the same as it was for the first release. The system is said to be optimized for up to 5000 users which results in that assumption clear box being checked of.

- *Coherent state*: Fulfilled – Due to that the university needs a better way of managing their administrative information it means that the requirements has been shared and the origin of them is clear, this also means that the rationale of the system is clear. Regarding conflicts it can be said to have been addressed as the faculty and management did not agree on how to deliver the students grades. Key usage scenarios has also been taken care of as this has been documented during the meetings with the users. The way this system makes management easier has been clearly understood as well as the importance of this system to prohibit problems.
- *Acceptable state*: Not fulfilled – There exists some minor complications with the system and requirements that's yet to be completed. It is not clear if the current state of the project can be said to have met all the important requirements.

Scenario 2 – Team Alpha

- *Seeded state*: Fulfilled – There are plans for developing a second release of the system. Possible constraints are defined to consists of problems with communication. There is a obvious need for team expansion as the system gets more features and requirements, this means that the growth mechanism checklist has been taken care of. Since the team consists of one person it's hard to state what kind of leadership that it's implemented. This also means that responsibilities and required knowledge is quite straight forward identified due to that there is no debate about which team member that is responsible for a specific task. Commitment is at a high level and work is going well. This results in that the seeded state has been fulfilled.
- *Formed state*: Fulfilled – Due to the structure of the team it's easy to see that the responsibilities are outlined depending on the compency of the developer. For this particular project there was enough developers but as the product/service expands there will be demand for more developers to meet the expectations. Developer is working good and accepting its goal that have to be met. External collaborators are not needed at the time of writing this. Communication between developer and stakeholder has been in written form.
- *Collaborating state*: Not fulfilled – Due to the structure of the team it's hard to argue that is has been a particular level of collaboration inside the team. However, one can argue if the communication between developer and stakeholder is to be counted as collaboration.

