Luleå University of Technology

Third year project

Sensor data aggregation through CoAP

Authors:

Sophia Bergendahl
sopber-8@student.ltu.se

Edvinn Bruun
edvbru-9@student.ltu.se

William Gustafsson
wilgus-9@student.ltu.se

Christoffer Holmstedt
cihhol-7@student.ltu.se

Marcus Rådman
marrdm-9@student.ltu.se

Kristoffer Svensson
kirsev-9@student.ltu.se

Ludwig Thurfjell
ludthu-7@student.ltu.se

Supervisors:

Ulf Bodin
ulf.bodin@ltu.se

Jens Eliasson
jens.eliasson@ltu.se

Rumen Kyusakov
rumen.kyusakov@ltu.se

March 2, 2012

Project Description

Background

Luleå University of Technology conducts research on lowpower wireless microprocessors called "Mulle". These microprocessors can be used for various things depending on which type of sensors you connect to it, everything from measuring temperature or vibrations in a car to analyzing the quality of the road that you drive on.

Every year northern parts of Sweden are used for testing cars during winter conditions. To test a car you first decide what you want to test, then you test with local sensors logging within the car. When enough data is collected you return back home. At the testing facility the data is now available for analysis. Depending on the results from the previous runs you might want to test some parts in more detail so you re-configure all sensors and go out for another test run.

This process is time consuming when you need to return to testing facility to be able to analyze and re-configure all sensors. In todays society most computers are connected to internet and/or other private networks, most of these computers have the ability to be remotely configured and maintained. The goal with this project is to be able to analyze data from sensors in realtime and re-configure them on the fly while testing is in progress.

Project Targets

1. Be able to send live sensor data from multiple "Mulles" to an online logging server/service.

2. Be able to read sensor data on the web with both a PC (web browser) and through an Android mobile device.

3. Be able to re-configure the sensors through a web interface and through an Android mobile device.

Technical dilimiations

TODO: Vad har explicit uteslutits från arbetet?

Execution of the project

Scrum and how it has been used

It was decided back in november that the entire project would be divded into three sprints. The exact dates were to be decided in the beginning of each sprint. In cooperation with the client the scope of the project and the scope of the first sprint was decided upon in november. During the first projectmeeting the first sprint goal was divided into eight sprint stories. It soon became clear that those eight stories were way to big, at the end of the sprint none of the stories had been finished.

Lesson learnt, the second sprint was divided into smaller stories which gave immidiate result when the first 69 sprint story points finished during the second sprint.

To decide upon size for each sprint story, for the second and third sprint, ”planning poker” [1, p. 42] was used. For every sprint story each project member everyone wrote down an estimate on the scope for each story. With planning poker it became clear that each project member had a different vision for each story. A short discussion after each estimate made it more clear on how big the scope was, an agreement was usually made within a few minutes.

One project, three sprint goals

TODO: Vi delade in oss i tre olika grupper, mulle, server och android under ett skype-möte. TODO: Efter sprint planning första två gångerna var det upp till varje ”grupp” att fördela uppgifter. TODO: Under andra sprinten flyttade vi resurser från server-delen då den var långt före de andra delarna. TODO: Flyttade tillbaka en resurs i början av tredje sprinten eftersom det fortfarande var svåra flaskhalsar vi satt fast med på Mullen...onödigt att tre sitter fast på samma ställe. TODO: Sista sprinten tog var och en direkt en uppgift från sprint backlog och satte den mer tydlig som ”sin”. Detta ledde till viss förbättring men eftersom vi inte har dagliga scrum möten så ger det inte en daglig uppdatering/reflektion över hur det faktiskt går...sprint backloggen blir inte en del av vardagen.

Individual time monitoring and speed

Sophia Bergendahl

Edvinn Bruun

William Gustafsson

Christoffer Holmstedt

What I've learnt during this project is that there is no way to estimate a reasonable time without knowing something about the topic of concern beforehand. I've had several stories assigned to me in the topic of linux installation and configuration e.g. installing a webserver and a mysql database. Both the webserver and database I've installed several times before and knew exactly which steps I was supposed to do to get it up and running as soon as possible. Of course this made it really easy to estimate an expected time for these stories, or at least most of them. Even though I have installed Ubuntu several times before I didn't anticipate that an entire installation could be so slow in the year of 2012, I had simply missed to take into account that we were running our server on a very old machine. The time estimate for this story was of by 100%.

Marcus Rådman

Kristoffer Svensson

Assignments

1. "Restructure the loading of new services for the python server"

   I had some previous experience with python on top of already been dealing with the actual server implemention we decided to use prior to dealing with this issue so I was pretty certain I would be able to complete it in about 3 hours. Without any major issues I managed to complete it in roughly 3 hours and 30 minutes. The question at that point was rather if the solution was adequate, which after some consultation it was deemed to be.

2. "Figure out how service discovery works in CoAP"

   The estimated time for this was set between 8 and 10 hours. Reason being that it was new ground for me but still didn't feel like it was that big of an issue. It turned out to be a relatively small fix after 6 hours of work. The issue of whether the solution was good enough or not for the end-purposes was brought up again but it was decided that it was good enough for the time being. With that said, I doubt that the estimated 8-10 hours would have been sufficient if a end-purpose qualifying solution was to be made.

3. ”Implement EXIficient”

   Originally the time for this assignment was set to 20 hours, when the assignment was still ”Implement EXI parser”. In that situation we assumed that there was more work to be done making the parser from the ground up.

   When EXIficient was discovered we found out that it had a ready-made demo that almost suited our needs. It turned out that the actual time needed to get something that would do the job for us would be around 10 hours modifying that demo, which is exactly what ended up happening.

4. ”Individual documention”

   I assumed about 2-4 hours for the actual writing with an added hour or two for the research needed to know what was to be included in the report. In the end, this all came down to about 4-5 hours which is well within the expected range.

General thoughts concerning time estimation I feel as if my personal time estimations, contrary to what I expected, have been quite accurate. I reckon the reason to this being that I had the fortune of having some kind of knowledge concerning the assignments prior to doing them. Without any idea on the scope or size of the assignments I think that the estimations wouldn’t have been this accurate.

The estimations themselves have been useful tools for knowing when to ask for help or guidance. When you’re approaching a time limit and you know that you’re kind of stuck that’s a really good indicator that help is needed. I think it’s been quite a useful tool, at least for me personally.

Ludwig Thurfjell

Reflection about Scrum usage during this project

Results

Deliverables

TODO: Vad levererar vi med respekt till ursprungliga krav?

Testing

Lessons learnt

Suggested improvements

Conclusions

References

[1] Henrik Kniberg, *Scrum and XP from the Trenches.* C4Media Inc, Publisher of InfoQ.com, 978-1-4303-2264-1, http://infoq.com/minibooks/scrum-xp-from-the-trenches, 2007.

Appendix A - How to build upon our codebase

This appendix include information on how to build upon our codebase for the Mulle (C), server code (Python, PHP/HTML5 and C) and Android Mobile phone (Java).

Mulle

Server

Coapy server

TODO: Python parts such as the python coapy server and how we use EXIP c-code parts.

Webpages and database

Android Mobile Phone application