

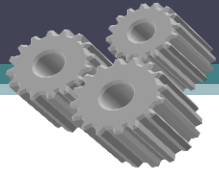
ESEM Industry Day

Technical Debt in a Mobile Application Development

Fabio Q. B. Silva, Claurirton A. Siebra, UFPE

In collaboration with Carolyn Seaman, UMBC and UFPE.





The Industrial Context

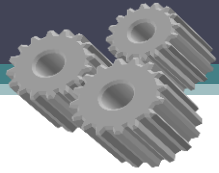


Industrial collaboration on
Application development and **Tests** for mobile devices

~ 80 developers

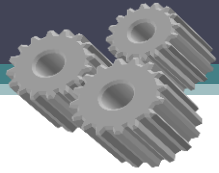
Since 2004





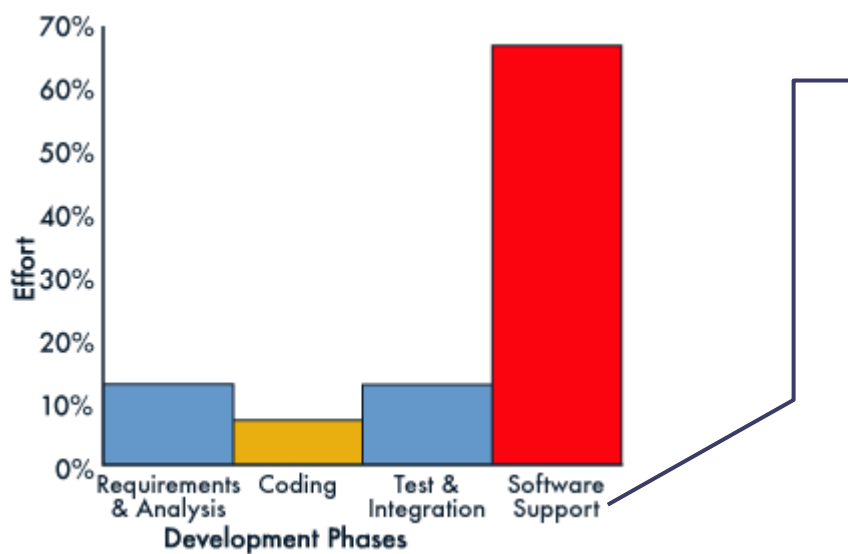
What is Technical Debt?

- It is a metaphor ...
- Short term quick-and-dirty **decisions** incur in **debt**
- .. that will have to be **payed** in the future
- .. with **interest**.



General Motivation for the Research on TD

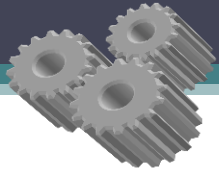
- Industry software evolves over time



Maintenance and Evolution

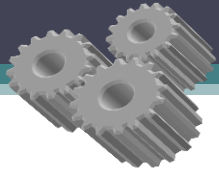
- Corrective Maintenance
- Adaptive Maintenance
- Perfective Maintenance

- Software needs to be developed in such way that supports its maintenance and evolution



General Motivation ...

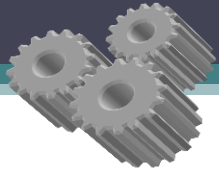
- Lehman: the loss of quality in software maintenance along the time
 - All software needs to be considered as an incomplete project, a dynamic system and in continuous evolution
- Most research has been focused on solutions to decrease the need for maintenance.
 - However this scenario is very rare, or ever impossible.
- New methods and techniques to predict impacts of the decision are fundamental to support the maintenance and evolution activities



Practical Motivation

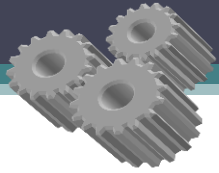
SMB (Samsung Mobile Business) system

- Mobile client for Microsoft Exchange™ e-mail
- Total of 63.218 logic code lines
- Maintained during 6 years (2005-2011)
- Translated to 17 languages and released in the European, South American and Asian markets



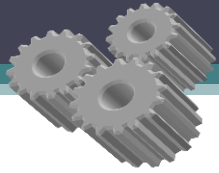
Practical Motivation

- We made a (necessary but bad) decision in SMB development
- Which incurred in (a lot of) technical debt
- We did not managed it (because we did not new it)
- ... and this caused 10 months of delay in the release of a crucial version of the system on the market



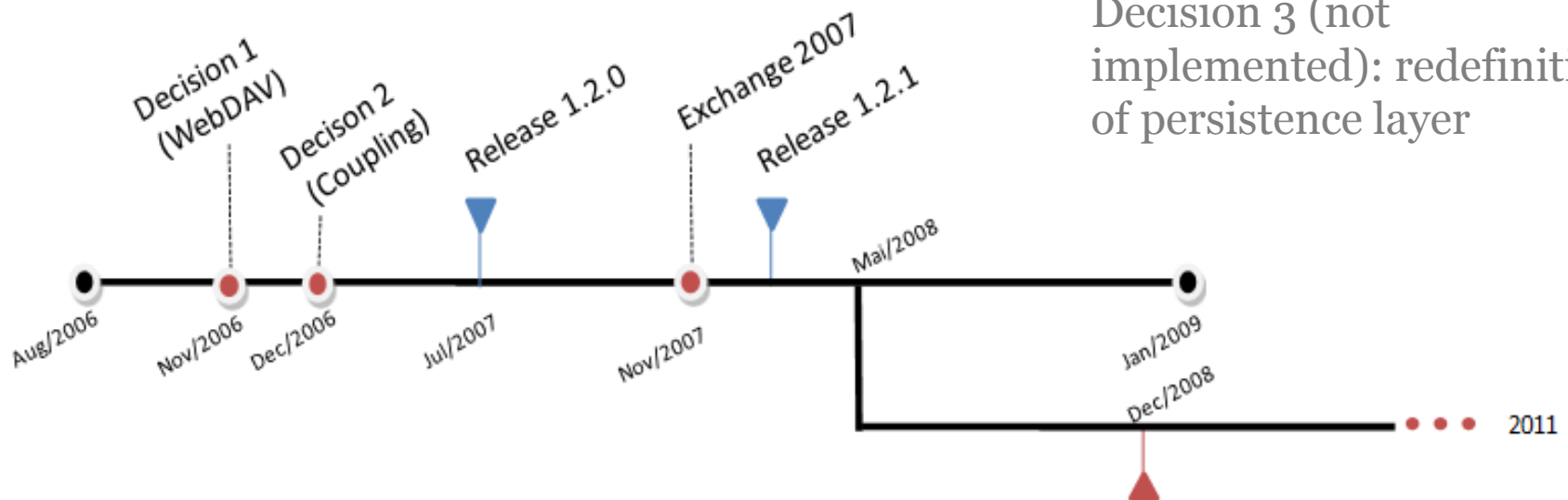
Why TD could had been useful?

- TD could be used as a way to manage and communicate long-term consequences of that (and other) decisions.
 - First, we could identify if such decision would incur in TD
 - Then, we should/could answer some question:
 - Which were the **benefits** of incurring in TD?
 - How much was the **debt** that we were accepting?
 - How we could measure the **interest**?
 - When and how much should we pay of the TD in the future?

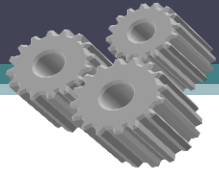


Our Approach to Understand the Problem

- Recreate the timeline evolution of scenarios
 - Related decisions were associated in scenarios for investigation

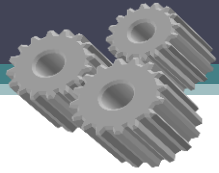


Sequence of decisions related to the protocol changing



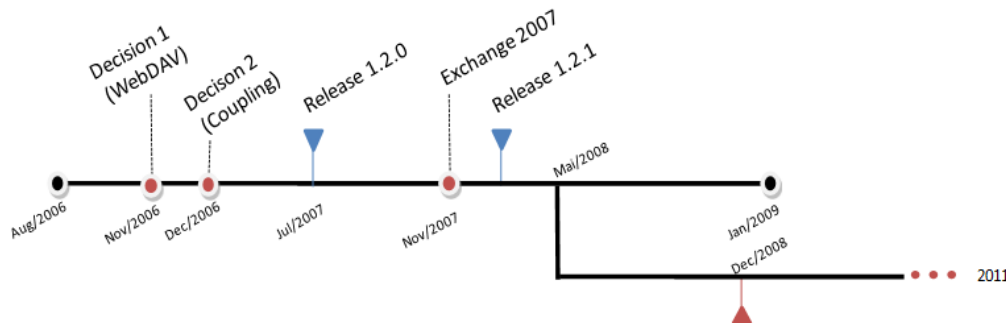
TD Characterization

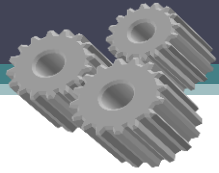
- After the analysis of this scenario, the following key TD aspects were characterized
 - Extra cost
 - Communication problems
 - Problem of accepting new requirements
 - Loss of debt visibility
 - Criticality of certain elements of the system



TD Characterization

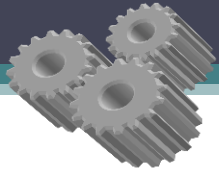
- Extra cost
 - Effort to implement Decision 3 at the moment of Decision 2 (estimation)
 - 345 h/person
 - Effort to implement Decision 3 after Decision 2 was implemented (real)
 - 1416 h/person
 - Total interested (actually paid)
 - 1071 h/person
 - Extra efforts to carry out the redefinition of changes (uncoupling communication and persistence layer)





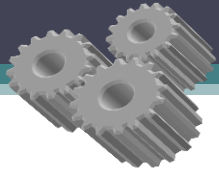
TD Characterization

- Communication Problem characterization
 - Period of six months between the decision of changing the protocol (Nov/2007) and implementing it (May/2008)
 - Problems in the negotiation of deadlines between technical and business teams
 - The communication problem occurred due to visualization problems of the debt interest by business team, at the moment of the debt payment
 - Evidences were detected in the emails analysis



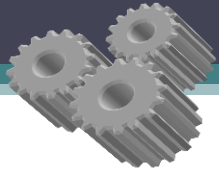
TD Characterization

- Problem to consider new requirements
 - 81% of the effort was used to pay interests related to the debt
 - This effort could be used to implement new features and add value to the application
- Real example:
 - Development of the touch screen technology, which was not implemented due to lack of time.
 - Touch screen support had been planned. However, it was put on standby because the protocol change became a higher priority
- Maybe the touch screen feature could have been a “killer” feature of the product



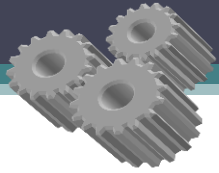
TD Characterization

- Loss of Debt Visibility
 - The identification of incompatibility, between the application and MS Exchange 2007, occurred before the first release of such application (Release 1.2.0).
 - During the planning phase of the Release 1.2.1, the team did not consider the alternative of paying the debt
 - Rather, in this release, the team implemented new functions and brainstorming was promoted to generate new ideas for this application



TD Characterization

- Critical elements of the system
 - One of the analysed TD item (communication protocol) affected the main function of the system (email server access to perform a download of the last updates)
 - This fact may have transformed the debt payment in a mandatory activity
- In other non critic elements of system, the obligation of paying the debt could be postponed
- The fact that a debt affects a key element of the system can influence the critical aspect of such debt and also the need of monitoring it



Conclusion

“A team that decides to compromise the quality of a system to attend a short-term demand rather than developing an appropriate long-term solution must completely understand and manage the risks or future consequences of its current decisions”

- TD provides the theoretical strengths to assist the characterization and monitoring of decisions
 - However we still have lacks in ways to identify, measure and manage TD aspects
- Samsung has invested in this direction and we intend to continue our research and use the acquired knowledge to assist the evolution of new projects