



Agile Software Development of
Embedded Systems

Version: 1.0
Date: 2006.05.10
Pages: 14

Authors

Santiago Estela

Status

Final

Confidentiality

Public

Agile Deliverable D.5.2.6

Experience Report on AgileReq Pilot Project

Abstract

The purpose of AgileReq pilot project: brief description of the work done and conclusions about the tool.

This document is part of the work of the EUREKA Σ! 2023 – ITEA 03003 AGILE project.

Copyright © 2003-2006 AGILE consortium.

Authors

Partner	Author	Email
SQS	Santiago Estela	sestela@sqs.es

Changelog

Date	Version	Editor	Description
2006.05.10	1.0	Santiago Estela	First version

Contents

INTRODUCTION.....	4
1 PROJECT DESCRIPTION.....	5
1.1 Agile approach	6
1.2 Challenges for the company	6
1.3 Objectives for SQS	6
2 RELEVANT METRICS.....	7
2.1 Effort related metrics	7
2.2 Error related metrics	7
2.3 Other metrics	8
3 GRAPHICS AND FIGURES	9
3.1 General information.....	9
3.2 Effort.....	10
3.3 Errors.....	11
3.4 Other graphics	13
4 CONCLUSIONS.....	14
4.1 Conclusions about the project.....	14
4.2 Conclusions about AgileREQ.....	14

Introduction

One of the most recurrent problems in the tailor-made developments is the little implication of the customers during the development process. Deficiencies in requirements (for example, incomplete or ambiguous requirements) are one of the biggest causes of failures in the software projects. Because of this key importance of a good requirements specification, it is essential to have a proper tool focused on the requirements engineering phase which makes requirements gathering, their estimation and their validation easier.

For this purpose SQS has designed AgileREQ. AgileREQ has been developed to help people involved in an agile project to go through the requirements definition phase achieving the best and fastest results.

This document describes the use of AgileREQ in a pilot project and presents the results obtained highlighting the advantages and the improvement obtained.

1 Project description

The project to be analysed was selected according to some criteria such as:

- Company's own previous experience
- Internal or external project
- Relation with the customer
- Probability of changing requirements
- Project length
- Size of the development team (small)

It is important to notice that this pilot is focussed on the tool so the project was selected to provide the most interesting trial possible. Because of this the following characteristics were chosen:

- The company did not have any previous experience in agile methodologies. This highlights the improvement in effort estimation.
- External project: to differentiate roles.
- The customer was accessible: It is essential to carry out a "pure" experience; otherwise results, such as concept errors, would not be real.
- Requirements definition was not in full detail: In order to be able to develop the Planning Game.
- Changes were likely to happen: It helps to highlight the improvement compared to classic processes.
- Short-medium project (6 months): Long enough to have an experience on the tool.
- A number of iterations could be planned (3 iterations). They are essential to check the evolution of estimation and errors and to include changes.
- Small development team (5 developers): big enough to put into practice some agile practices.

Prior to any work in the project, all the people involved received an initial training on Agile Methodologies, consisting on the explanation of different agile methodologies, their way of implementation and benefits, with an special emphasis on XP (since this is the most popular and extended agile methodology), and on the requirements engineering phase.

1.1 Agile approach

The process was mainly based on XP. Most of its practices were applied emphasizing those related to requirements such as:

- Planning game.
- Metaphor. That means common understanding of the system.
- On site customer.
- Acceptance testing.

As for the practices not having been applied:

- Unit testing was not applied because of the need of automated unit tests and the great effort that this practice demands.
- Refactoring can not be even considered without the support of a suite of automated unit tests.

The requirements were gathered as User Stories and these US were divided in Tasks being assigned an Effort (both in days and in hours) and a Developer.

User Stories were selected to be developed and the Iteration Planning was set at the beginning of each Iteration.

When changes appeared, the subsequent modifications to the Iteration Planning were made and the Deviation Due To Changes value was estimated.

1.2 Challenges for the company

Apart from the general and typical objectives, like time and costs reduction and quality improvement, the concrete challenges within the company were:

- To identify more accurately where and when errors appear.
- To improve the effort estimation and avoid deviation.
- To reuse work
- To gain experience with agile methodologies.

AgileREQ has proved to be a very useful tool to reach these objectives.

1.3 Objectives for SQS

The main objective for SQS was to measure how adequate AgileREQ was for such kind of projects and how helpful it was to improve quality, accuracy and effort.

- Quality measured as lack of errors. Both development and concept errors.
- Accuracy related to effort estimation but also to requirements definition.
- Effort optimization by avoiding error-solving effort.

2 Relevant metrics

AgileREQ has been designed to gather the most useful information about metrics with the minimum impact in work load.

By introducing some key data during the process (such as Developer ID, efforts, errors etc), users can obtain figures and graphs that have been proved to be really useful to identify inefficiencies and wrong strategies.

The most relevant metrics are grouped in two categories (Efforts and Errors) but there are many others. Let's see a brief explanation of the metrics.

2.1 Effort related metrics

There are two main groups of Effort metrics:

- Estimated Effort and
- Real Effort.

For each one of the defined tasks, an Estimated Effort is assigned and a Real Effort is measured. Taking advantage of this data combined with other information such as developer, user story, iteration or date, many figures and graphs are provided including data about deviation between Estimated Effort and Real Effort.

Some interesting metrics are:

- Effort Deviation / Iteration
- Effort Deviation / User Story
- Effort Deviation / Developer
- Real Effort / (Day and User Story)
- Real Effort / Iteration

During the pilot project, the necessity of a new concept was identified and the variable "Effort Deviation Estimation due to Changes" was included.

It is specified when changes appear and new estimations must be done. It is an estimation of effort deviation caused by the changes. Total Effort Deviation can be calculated including or not this variable.

2.2 Error related metrics

There are two main groups of Error metrics:

- Development Errors and
- Concept Errors

Every time an error is detected, it is recorded, categorised and weight (negligible, marginal, serious, critical and catastrophic). Once more, depending on developer, user story, iteration and even date, many figures and graphs are provided.

Some interesting metrics are:

- Concept Errors / Iteration
- Concept Errors / User Story
- Critical, Serious, Marginal Errors / Date
- Critical, Serious Development Errors / Developer

Changes can be shown in the graphs by including the value of “Effort Deviation Estimation due to Changes” which gives an idea of the importance of the changes.

2.3 Other metrics

Some other metrics and the combination of metrics of both groups can be generated:

- Number of Validated User Stories
- Number of No Validated User Stories
- Number of Iterations and Tasks
- Number of Developers
- ...

Combinations:

- Effort Deviation / (Concept Errors and Iteration)
- Effort Deviation / (Development Errors and Iteration)
- Iterations / Days
- Validated User Stories / Iteration
- ...

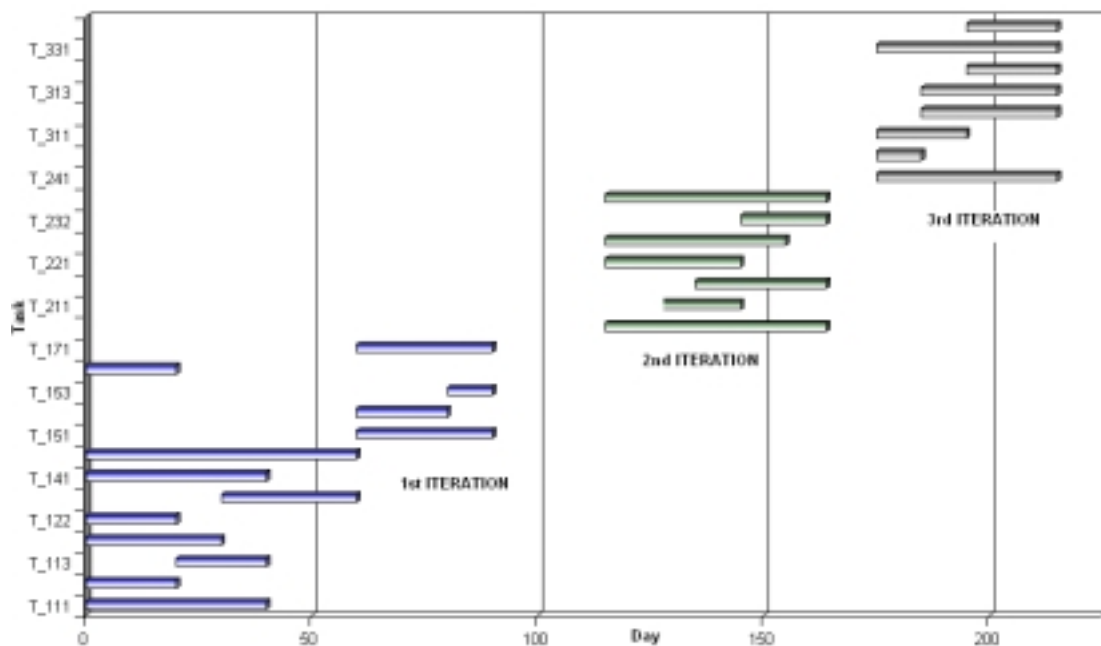
3 Graphics and Figures

During the development of the project, a lot of information was gathered. In this section, only the most relevant information is provided.

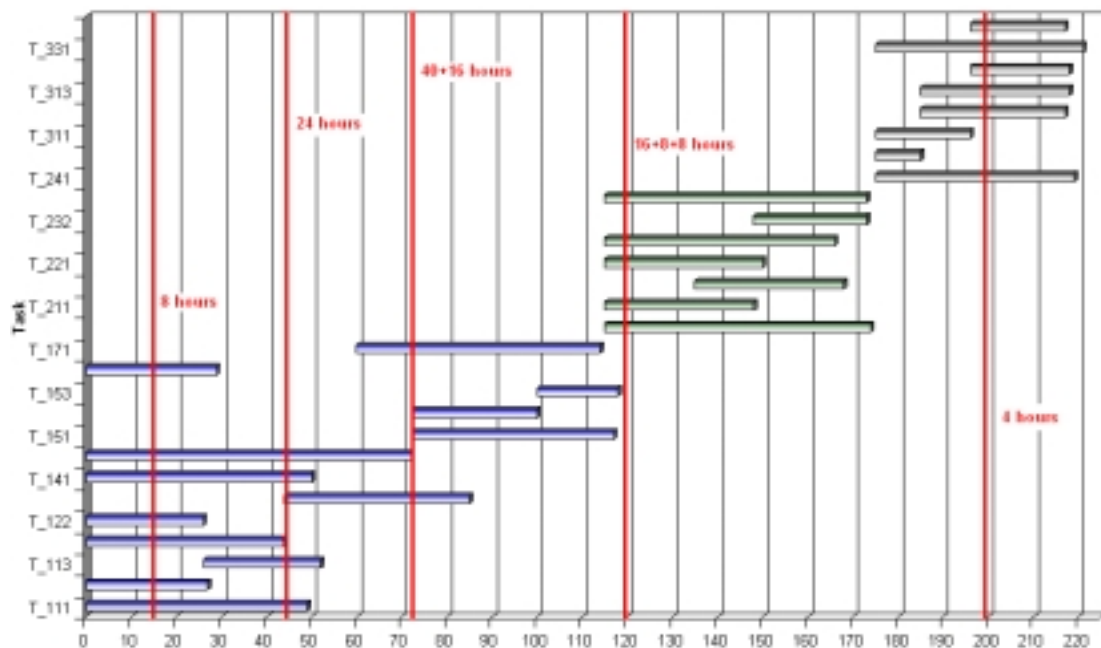
3.1 General information

The project was planned to be developed in 6 months but the real length of the project was over 7 months.

In the following graphs Estimated and Real Lengths are shown.

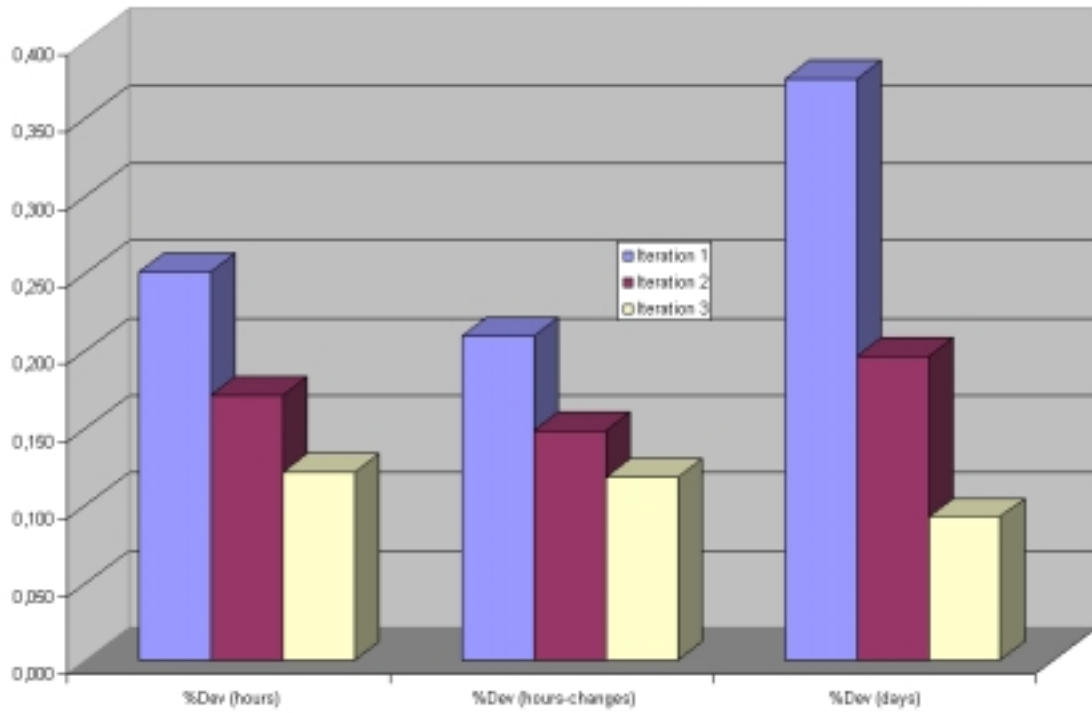


F 3.1 Iteration planning

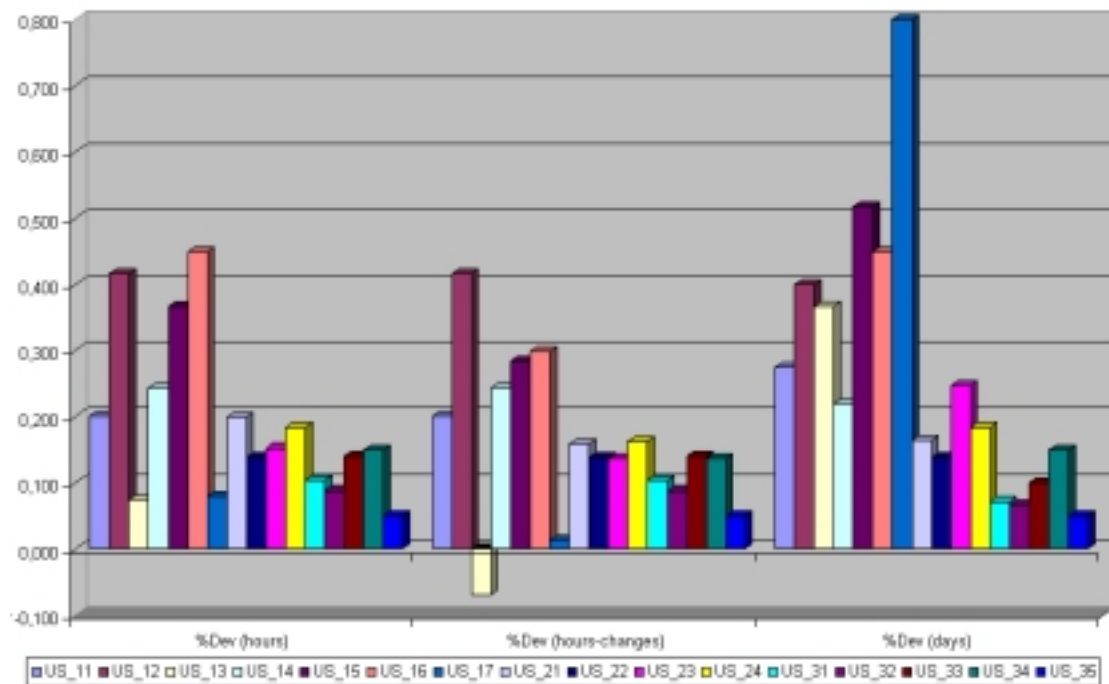


F 3.2 Real Length + Changes

3.2 Effort

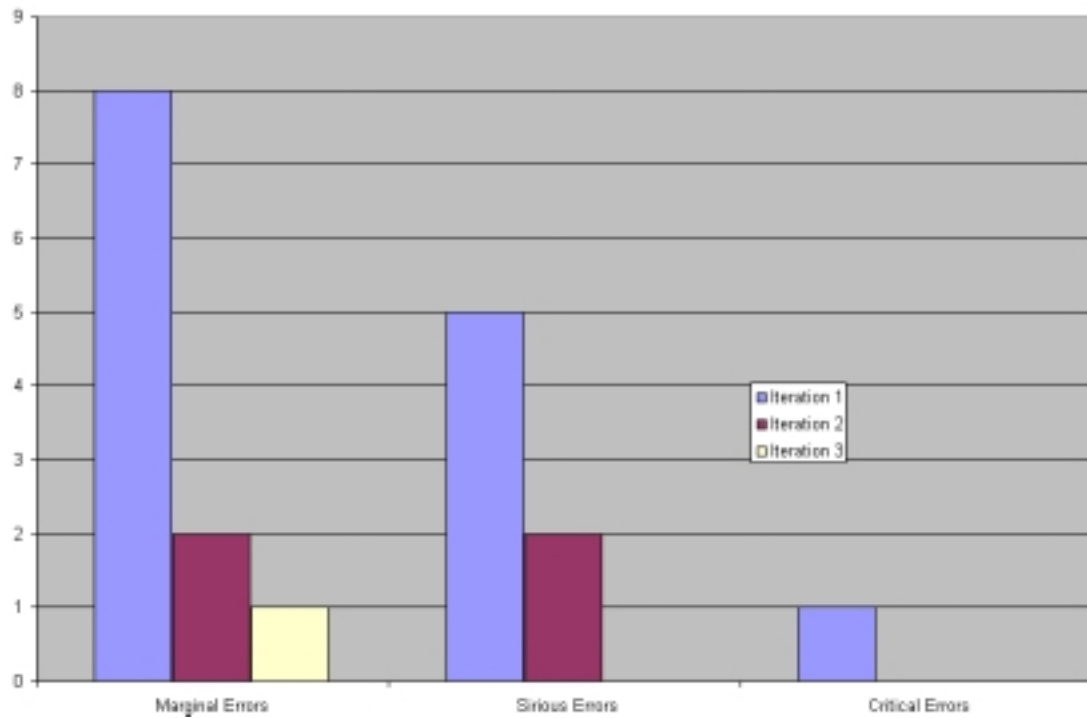


F 3.3 Deviation/Iteration

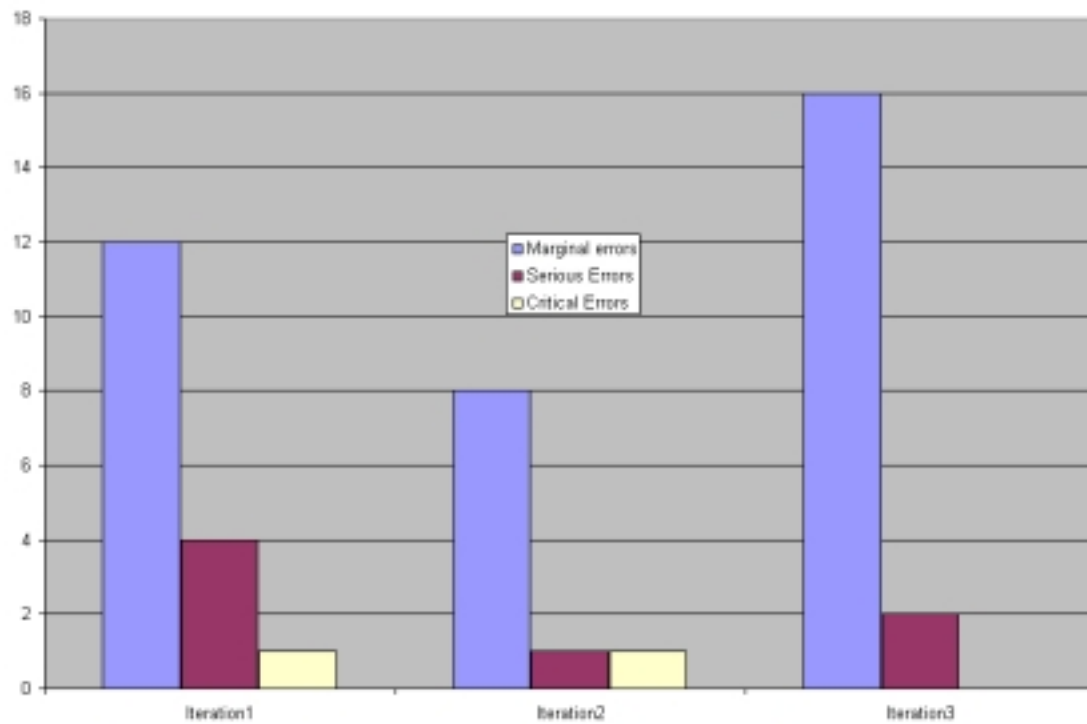


F 3.4 Deviation/US

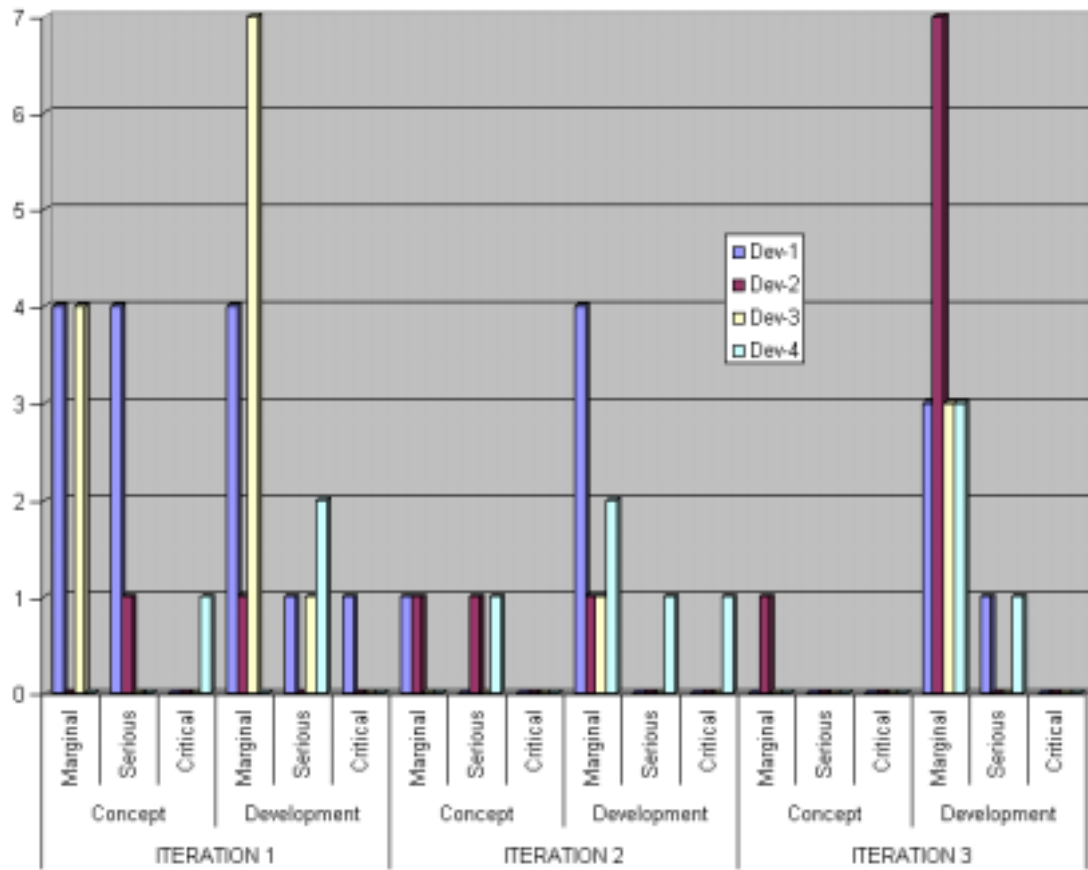
3.3 Errors



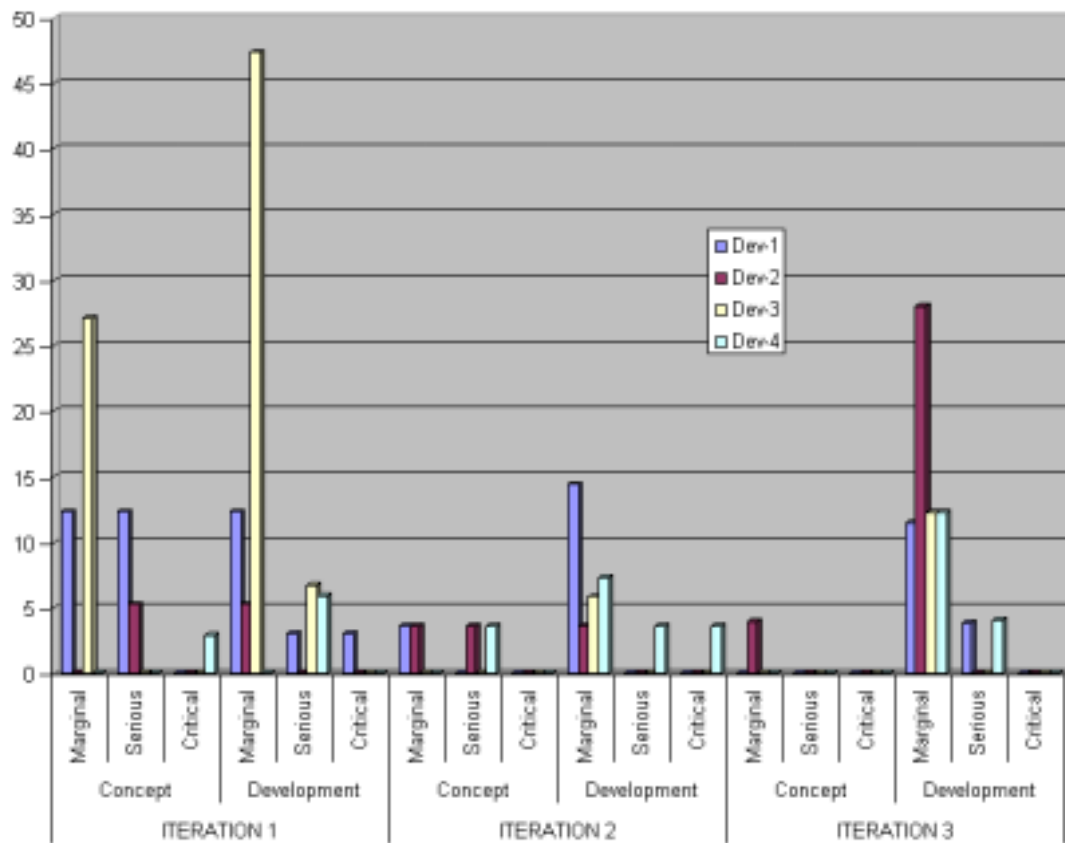
F 3.5 Concept Errors



F 3.6 Development Errors

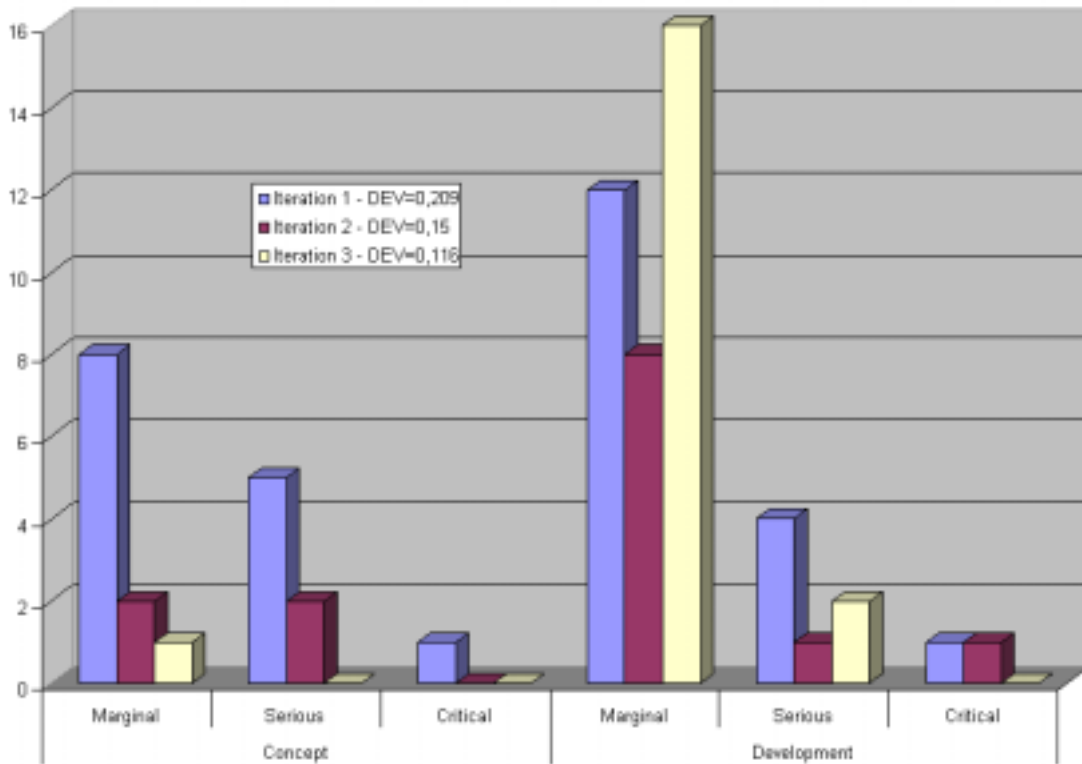


F 3.7 Errors/Developer



F 3.8 Error RATE/Developer

3.4 Other graphics



F 3.9 Errors ↔ Effort deviation

4 Conclusions

Conclusions can be divided in to sections, conclusions about the project and conclusions about AgileREQ. As this document is focused on the tool, the first section is only mentioned but not explained in detail.

4.1 Conclusions about the project

The little concreteness on the description of the requirements was a handicap in this project, however, applying XP practices helped during the process. Although there were differences between planned and real effort, dramatic deviations were avoided both in planning and in results and it was possible to improve effort estimation in every iteration thanks, partly, to the analysis of the information gathered.

Concept errors were earlier detected thanks to the continuous contact with the customer and the acceptance testing.

Changes (closely related to concept errors) were easier accepted.

4.2 Conclusions about AgileREQ

The most useful data is compiled throughout the process with the minimum effort.

Being a web access tool allows the centralization of the information about remaining work, project status, changes, results, errors...for all the components of the team, including the customer.

Errors, mistakes and team misfunctions can be earlier detected and more cheaply solved.

AgileREQ has proved to be an intuitive and easy to use helpful tool, however some aspects have to be remarked.

- Team members must be methodical. Although not much information is required, it must be provided accurately and punctually in order to obtain valuable results. Otherwise using AgileREQ is a waste of time.
- With regard to reports, there is still work to be done. Some reports are automatically generated and data can be exported, but reports are not yet customizable and data structure is sometimes hard to be used.