



Agile Software Development of Embedded Systems

Version : 1.0
Date : 2005.02.09
Pages : 8

Authors

Jesús M. Sagredo

Status

Final

Confidentiality

Public

Agile Deliverable D.4.3

Setting up the Criteria for the Deployment Model

Abstract

The objective of this document is to set up some criteria in order to deploy the Agile software development process of embedded systems within the industrial partners.

This document is addressed to the project managers and software developers of the company.



I T E A

INFORMATION TECHNOLOGY
FOR EUROPEAN ADVANCEMENT



TABLE OF CONTENTS

CHANGE LOG	2
APPLICABLE DOCUMENT LIST	2
EXECUTIVE SUMMARY	3
1. Introduction.....	3
2. Questionnaire	4
2.1 General information	4
2.2 Organizational profile.....	4
2.3 Objectives and Constraints	4
2.4 State	5
2.5 Suggestions	7



CHANGE LOG

Vers.	Date	Author	Description
0.1	2004.12.13	JMS - SQS	First Draft
1.0	2005.02.09	JMS - SQS	Add objective of the questionnaire and who should fill it

APPLICABLE DOCUMENT LIST

Ref.	Title, author, source, date, status	Identification

	AGILE document template Deliverable ID: D.4.3	Page : 3 of 8
		Version: 1.0 Date : 2005.02.09
		Status : Final Confid : Public

EXECUTIVE SUMMARY

1. INTRODUCTION

This document is to set up the criteria for the deployment model that is the information needed by the industry.

A deployment can be generally defined as a systematic process, through which an activity or process is introduced to all applicable areas of an organization

In the software-engineering process, the purpose of the deployment is to ensure a successful transition of the developed process to its users.

The objective of this document is to set up some criteria in order to deploy the Agile software development process of embedded systems within the industrial partners.

This document is addressed to the project managers and software developers of the company.



2. QUESTIONNAIRE

2.1 GENERAL INFORMATION

Organization: _____

Department: _____

Address: _____

Name of the respondent: _____

E-mail: _____

Official title & job description: _____

Respondents position in organization: Organizational level
 Project level

2.2 ORGANIZATIONAL PROFILE

Area of business: _____

Size of organization/units:

Total amount of employees < 50 50-500 > 500
Employees in R&D < 50 50-200 > 200
Employees in Software Development < 20 20-100 > 100

2.3 OBJECTIVES AND CONSTRAINTS

	Objective/Constraint	Priority (1 low, 5 high)
C1	Decrease time required to test	
C2	Decrease cost of testing	
C3	Increase the quality of testing (and systems)	
C4	Minimise change to current practice	
C5	Get a quick payback	



2.4 STATE

	<i>State Description</i>	<i>Low (score 1)</i>	<i>Medium (score 3)</i>	<i>High (score 5)</i>	<i>Score (1-poor 5-good)</i>
S1	Testing has no deliverables of value	Testing deliverables are not identified, not valued, not used for anything	Test deliverables are used to make product release decisions	Test deliverables are used to make product release and process improvement decisions	
S2	Testing has little purpose	Testing is essential to achieve sign-off	Testing is essential to detect faults in software	Testing is essential to both detect and prevent faults in software	
S3	Test objectives are less important than development objectives	When development slips, testing time is squeezed to achieve deadlines	When development slips, testing time is squeezed, but extra resources are applied to achieve the deadline.	When development slips, a risk assessment is conducted and a decision to squeeze, maintain or extend test time may be made.	
S4	Testing is expensive, but the costs are not visible	Test activities are not identified or distinguished separately from development activities so costs are not visible	Only system and acceptance test activities are identified and tracked and costs recorded	Static tests (reviews, inspections etc.) and all dynamic test stages (unit, integration and system and acceptance tests) are tracked and costs recorded	
S5	Of the errors that are found, there is a perception (based on evidence) that many should have been found in earlier test stage(s).	Errors are found in acceptance tests that should have been found in sub-system and system tests.	Errors are found in system tests that should have been found in sub-system tests.	Errors found would not be expected to have been detected earlier	
S6	Users perceive that they are responsible for testing	Users compensate for poor development and system testing by staging large acceptance tests; developers and system testers regard user testing as a fallback	User and system testing is not coordinated; users insist on large scale tests	System and user testing is coordinated; users contribute to system test planning and prepare their own tests to complement system tests	
S7	Significant requirements and design defects are found by system and acceptance testers	Requirements documents are used 'for guidance only'. Acceptance testers regularly find defects so serious that they cause project slippage or force de-scoping decisions.	System testers have difficulty deriving tests from specifications. Test planning and the tests themselves reveal missing and ambiguous requirements that require late reanalysis or redesign.	System and acceptance tests reveal subtle or minor requirements or design defects. Defects found usually require re-coding, rarely reanalysis or redesign.	
S8	Lack of ownership of the test process	No one owns testing, it is assumed to be done by developers (who don't want it)	Developers, system testers and users own their test stages, but work independently	Overall approach to testing developed and agreed by consensus - developers, testers and business users and project management are involved in master test planning	



AGILE document template
Deliverable ID: D.4.3

Page : 6 of 8

Version: 1.0
Date : 2005.02.09

Status : Final
Confid : Public

	<i>State Description</i>	<i>Low (score 1)</i>	<i>Medium (score 3)</i>	<i>High (score 5)</i>	<i>Score (1-poor 5-good)</i>
S9	Users won't buy-in to the test approach	Users only become involved when the time comes to acceptance test	Users contribute to system as well as acceptance test planning	Users are fully involved in the preparation of the test approach, system testing, user testing and have visibility of test activities	
S10	It is difficult to get volunteers to test, to hire and retain testers	Testing is perceived to be easy -anyone can do it; testing is not recognized or rewarded; training is not given; management favour developers over testers	There are some career testers, some have been trained; testing is recognized, but not rewarded	Testing is seen as a viable career path; testers are promoted to management positions and are rewarded	
S11	There are gaps in the testing - features of the system may be released untested	Tests are not based on requirements or design documents, there are no test inventories or means of measuring coverage against requirements or specifications	Test inventories are used to define the scope of system and acceptance tests and cross-reference requirements; formal test techniques are sometimes used to design black-box test cases.	Test inventories are use for all testing and are reviewed against requirements and specifications, formal test techniques are used for test case design, tools are used to measure code coverage	
S12	There are gaps in the testing - testing does not address user concerns	System testing does not take account of user needs and usage patterns, user testing is ad-hoc and disorganized	Some system testing is based on the business process, user tests cover variations in data but not business process	System tests are based on business transactions and cover business process paths, user testing focuses on usability and fit with the business process	
S13	There are gaps in the testing - some non-functional requirements of the system may not be tested	Non-functional requirements and concerns are not documented, non-functional tests are not performed	Some informal non-functional tests (usually performance) are performed but are squeezed and often left incomplete	Non-functional requirements are documented and used to steer testing; non-functional tests are planned early and given enough time in the project schedule to be meaningful	
S14	There is no evidence available to show that testing has been thorough	Test plans do not set measurable targets, test records do not provide an insight into thoroughness	Test inventories are used to define quantifiable targets. Test design techniques are sometimes used.	Functional test techniques are consistently used to design tests. Source code analysis tools are used to measure structural coverage.	
S15	Test budgets are unreliable because there is no rational method of estimating test activities	Test budgets are calculated as a percentage of development budgets	Test budgets are estimated, based on 'received wisdom' and experience	Test budgets are estimated based on previous project metrics and an assessment of the risks to this particular project	



AGILE document template
Deliverable ID: D.4.3

Page : 7 of 8

Version: 1.0
Date : 2005.02.09

Status : Final
Confid : Public

	<i>State Description</i>	<i>Low (score 1)</i>	<i>Medium (score 3)</i>	<i>High (score 5)</i>	<i>Score (1-poor 5-good)</i>
S16	When testing stops, no one knows whether enough or too much testing has been done.	Testing usually stops when time runs out. Some tests may never be run.	Testing is time-limited, prioritization happens after estimates and timescales are committed to	There is a defined prioritization policy. Tests are prioritised before test plans are fixed. Tests that are de-scoped and the associated risks are identified and understood.	
S17	Decisions to release into production are based on 'gut-feel' and not an objective assessment of the quality and readiness of the software	Test records provide scant details of the stability of the features of the system. Testers argue against release, but cannot provide objective evidence.	Testers log and categorize incidents. Features of the software can be assessed in terms of defects found, deferred, fixed/re-tested and outstanding. Little or no regression testing done on final version of software.	Defects are analyzed for severity, location. Assessment of the stability of all system features can be made. Regression tests on final version of the software have been run without incident.	
S18	Test environments are rarely adequate and never ready in time	Test environments are created just before testing needs to start	Test environments are requested and built in good time, but testing time is lost through lack of control, support resources and environmental instability.	Test environments are available in time for testing, are adequately supported and testing is rarely held up through environmental problems.	
S19	When software is first delivered into test, there is a delay of days or weeks before the first tests pass successfully	Testers complain that early builds are incomplete, unstable and not ready for testing. Testing time is lost through unreliable builds and test environment. No tests pass first time because of 'fundamental' software defects.	Testers complain that the configuration of the test environment causes major problems. Testing time is lost through the software behaving differently in test than in development. Early tests fail because of configuration discrepancies.	Testers are able to run tests on the day the software is installed in the test environment. The software is testable, insofar as it is stable enough to be usable on day 1. Incidents raised reflect functional errors in the software and not unreliable build	
S20	Testing is not focused, systematic or effective	Testers never know whether enough testing has been performed because they rely on gut feel. Testers are uncomfortable that they may be blamed for bugs in production.	Test documentation is produced in large volume; the simplest test design techniques are sometimes used. Much effort is spent testing more complex functionality but it is never tested 'thoroughly'.	Testers focus their attention on the key risks and select test strategies specific to the risk and type of software under test. Testers produce objective and quantifiable test coverage targets and tests consistently meet them.	

2.5 SUGGESTIONS

Please, add in the space below any objective that is not listed.

Please let us know any problem that is not reflected in the questions above, and it can be added to the State list.