Change Management

Cours LPSIL 2013

Questions?

Agenda

- Introduction
- Changes Types
- Risk Management
- Traceability
- Tooling
- Maintenance and Support
- Archiving
- Conclusion

Agenda

- Introduction
- Changes Types
- Risk Management
- Traceability
- Tooling
- Maintenance and Support
- Archiving
- Conclusion

Why changing ?

- Reasons of changes
 - New requirements
 - Improvement / Performance / Usability
 - Bug fixing
 - Feature request
 - New regulation
 - Customer request
 - Better maintainability (reduce costs)
 - 0

What can be changed?

- Almost everything!
 - Project scope
 - Schedule
 - Budget
 - Product content (features)
 - Environment / Platforms / OS / Language
 - Hardware
 - Documentation
 - Process
 - Customer contract
 - Market
 - 0

Why do we need to manage changes?

- Why do we need to manage changes?
 - Evaluate the rationale for the change
 - Assess risks & impacts
 - Estimate costs
 - Ensure all stakeholders will be informed/consulted
 - Get approval for the change
 - Plan changes
 - Log changes
 - Build knowledge
 - Track implementation status
 - 0

Agenda

- Introduction
- Changes Types
- Risk Management
- Traceability
- Tooling
- Maintenance and Support
- Archiving
- Conclusion

- Changes can be related to:
 - the project/product specifications
 - the organization
 - the environment

In the product specifications?

- In the product specifications
 - Requirements
 - New features
 - Change requests
 - Bug fix requests
 - List of supported platforms

In the organization?

- In the organization
 - People turnover
 - Merger / Acquisition / Reorganization
 - Processes / Policies
 - Management line
 - Business objectives
 - 0

In the environment?

- In the environment
 - Market target
 - Competitors / challengers
 - Regulations
 - Economic downturn
 - 0

- All at the same time?
- Impact on the development plan?

Example

- Iterative development process
- Planning phase (define baseline)
- Development iterations (track changes)
- Stabilization iterations (monitor changes)
- Code freeze milestone
- Delivery/Release phase (strictely control changes)

Agenda

- Introduction
- Changes Types
- Risk Management
- Traceability
- Tooling
- Maintenance and Support
- Archiving
- Conclusion

What is a risk?

- Risk Definition
 - An uncertain event or condition that, if occurs, has a positive or negative effect on a project objective

- How to identify risks?
 - Documentation reviews
 - Knowledge / Historical data / Lesson learnt of previous experiences
 - Brainstorming with key people
 - Interviewing / Consulting experts

- Risk Analysis
 - Probability of occurrence
 - Impact assessment
 - → Matrix

	Very likely	Medium 2	High 3	Extreme 5
	Likely	Low 1	Medium 2	High 3
	Unlikely	Low 1	Low 1	Medium 2
	What is the chance it will	Minor	Moderate	Major
	happen?		Impact	

- Risk responses ?
 - Avoid
 - Transfer
 - Mitigate
 - Accept

- Positive risks?
- Opportunities
- Strategies
 - Exploit
 - Share

Agenda

- Introduction
- Changes Types
- Risk Management
- Traceability
- Tooling
- Maintenance and Support
- Archiving
- Conclusion

- No change can be done without a (good) REASON
 - Defect (minor / major / blocker ?)
 - Planned item
 - Improvement needed
 - Code refactoring for better performance
 - Request from a key customer
 - Legal issue, copyright, licensing...
 - Risk of Revenue loss

- No change can be done without an APPROVAL
- Must be approved by all key stakeholders
 - Notify impacted people
 - Get their estimate/feeling
 - Get their GO formally
 - Summarize outcome and Communicate about the change decision

- Changes must be planned
 - What to change?
 - How ?
 - When?
 - Who must validate?
 - When do we consider change request is closed?

- Changes follow-up
 - Track the status of the change
 - Validate the change is conform
 - Write lessons learnt outcome for future projects
 - why this change?
 - how has it been implemented?
 - how much did it cost?

Agenda

- Introduction
- Changes Types
- Risk Management
- Traceability
- Tooling
- Maintenance and Support
- Archiving
- Conclusion

Tooling

- What tools in Software Development discipline?
- Which features expected?

Tooling

- Project Management
- Source Code management
- Defect Tracking
- Quality Management
- Documentation / Processes
- Customer case management
- Public website
- Remember the continuous integration

Tooling

- Project Management
 - Consult / Inform stakeholders
 - Approval / Go-NoGo

- Source code control
 - Revision number
 - Comment
 - ID + link to change request or fixed defect

- Issues Tracking
- Jira / Bugzilla or equivalent
- Can work for lot of change types
 - Bugs/Defects
 - Improvements
 - Wishes
 - Customer change requests
 - User story
 - Feature
 - Task

- Issues Tracking System
- What's an issue?
 - An ID for the change
 - A Status (Open / In Progress / Resolved / Closed)
 - A Priority (Low / Medium / High)
 - A Severity (Low / High / Critical / Blocker)

- Issues Tracking System
- What's an issue?
 - Reported by someone (created by)
 - In the hands of someone (assignee/owner)
 - Planned in a delivery (next release? fixpack? hotfix? patch?)

- Issues Tracking System
- What's an issue?
 - A knowledge base
 - Comment Thread
 - Link to Code changes (diff introduced)
 - Link to Build execution results
 - Link to Tests execution results

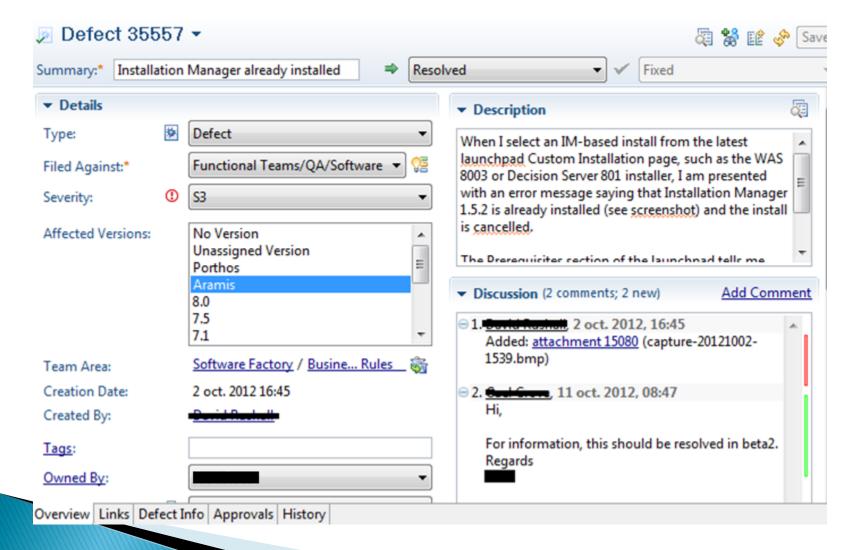
- Acceptance of the change ?
 - The developer commits the change
 - The continuous integration process operates
 - The build and the tests are executed

- Acceptance of the change ?
 - The developer marks the issue as « resolved »
 - The QA verify the fix and put the issue as « closed »
 - (can also be reopened if not fixed)
- Now ready for delivery to customer

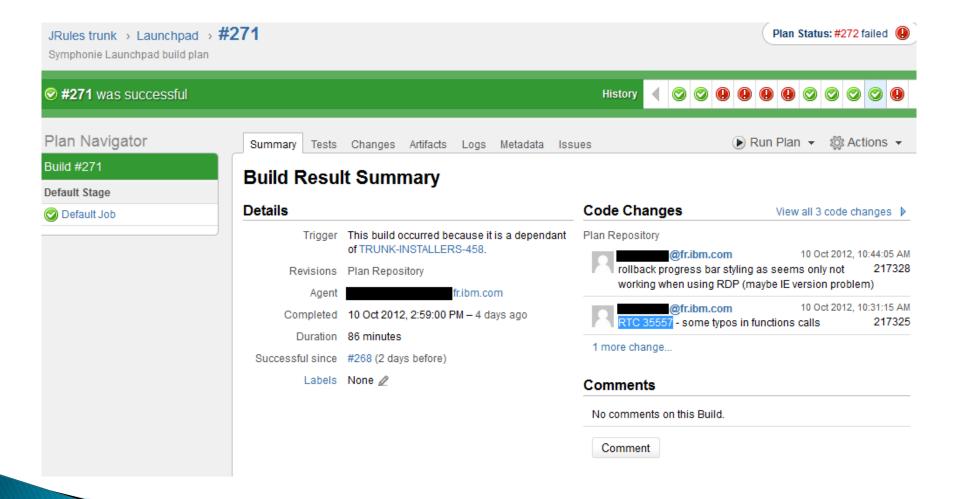
- Continuous integration ensures
 - Changes Traceability
 - Code change revision
 - Build validation / numbering
 - Test execution / reports
 - List of changes in each deliverable
 - Continuous validation of changes

- An example
- See the pictures
- The build will provide lots of valuable information
 - The build information itself (ID / Result)
 - The changes included + code revision numbers
 - The defect ID

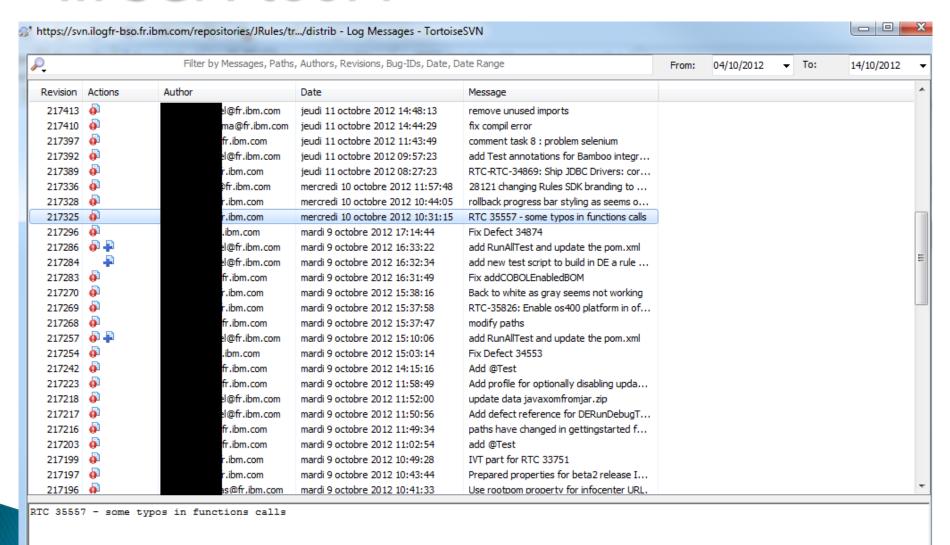
In Defect tracking system?



In the build tool?



In SCM tool?



In SCM tool - more details?

```
revisions 217324-217325, distrib - TortoiseUDiff
File
    1 Index: installers/IM/common/launchpad/com.ibm.wodm.launchpad/launchpadCont
    3 --- installers/IM/common/launchpad/com.ibm.wodm.launchpad/launchpadContent
    4 +++ installers/IM/common/launchpad/com.ibm.wodm.launchpad/launchpadContent
    5@@ -258,7 +258,7 @@
         if (top.OSTYPE == "windows") {
              return getUserWindowsInstallationManagerProperties(securityCheck);
    8
         } else if (top.OSTYPE == "unix") {
    9 -
              return getUserAdminUnixInstallManagerProperties(securityCheck);
  10 +
              return getUserUnixInstallManagerProperties(securityCheck);
  11
  12
  13 }
   14
```

Agenda

- Introduction
- Changes Types
- Risk Management
- Traceability
- Tooling
- Maintenance and Support
- Archiving
- Conclusion

How to manage the changes requested by customers?

- How to manage the changes requested by customers?
 - Contract
 - Communication with customer
 - Fixpack rollout strategy
 - Urgent issue management
 - Compatibility and migration

- Contract
 - SLA
 - Warrantee
 - Maintenance cost

- Communication with customer
 - Support engineers
 - FAQ
 - Hotline
 - • •

- Fixpack rollout strategy
 - End of service (lifecycle)
 - Frequency

- Urgent issues management
 - Hotfix
 - Critsit
 - Escalation process
 - Sending consultant onsite

- Compatibility and migration
 - Data migration
 - Plug'n'play
 - Public API

Agenda

- Introduction
- Changes Types
- Risk Management
- Traceability
- Tooling
- Maintenance and Support
- Archiving
- Conclusion

- What?
- When?
- Where ?

- What ?
 - Deliverables to customers
 - Source code
 - Build configurations
 - Documentation
 - Test plans

- When?
 - Each release?
 - Reccurent: weekly, monthly, etc...

- Where ?
 - Offsite
 - Third party

Agenda

- Introduction
- Changes Types
- Risk Management
- Traceability
- Tooling
- Maintenance and Support
- Archiving
- Conclusion

Conclusion

- Change must be assessed
- Change must be approved
- Change must be planned
- Change must be tracked/monitored/controled depending on the phase
- Change must be verified (closed! = DONE)

Questions?

