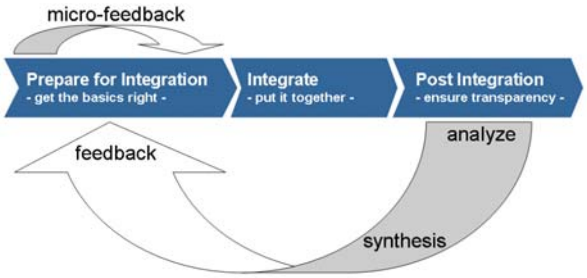
Efficient and Effective Software System Integration
Going Beyond
“Continuous Integration”

SE 2008 Tutorial Content, 19. 2. 2008

This tutorial covers one of the key problems in software development which is to know where you are today, especially in the light of complex software integration scenarios.

Without knowing what is really available – today - it will be very difficult, partly impossible to adjust a project plan accordingly. And project plan adjustment is essential to become successful in the future.

This tutorial is divided into three phases, outlined in the figure below:



The first phase – *Prepare for Integration* - of the tutorial addresses issues form the area software architecture what needs to be known upfront to be more successful later in the software system integration. Furthermore the tutorial takes as well a closer look into a developer work area, what are the typical problems of a developer and outlines possible solution scenarios. In this phase also issues of “How to integrate newbies?”, or How do code reviews more effectively?” of a development environment are taken into account and solutions will be presented

Phase number two – *Integrate* - outlines a principle called “Multi-Stage-Integration” which is technique for slicing software system integrations into smaller and better manageable pieces. This allows the easy detection of successful or not-successful integration paths.

Phase number three – *Post Integration* - combines different data sources available within a development environment for achieving a high degree of transparency. A new – not yet published - concept called “Unified Release Notes” will be used for this, the concept in a nutshell:

The Unified Release Notes (URN) concept provides a higher-level view of activities happening within a development organization. In combination with other tools, like Bugzilla CIA - the Open Source Informant and Trac, software development becomes very transparent. And nice. And functional. [1]

The URN uses data that is inherent in the change-management system(s); this data is deepened by structured, detailed input from the developers. The overview that URN provides increases transparency, tracking and overall understanding of activities, leading to more successful project management and better quality production.

Furthermore the URN provides a data foundation to apply research results such as the concepts for “Predicting Faults from Cached History[1]“. Furthermore, “The Future of Programming Environments: Integration, Synergy, and Assistance[2]“, outlines possible future programming environments, where this tutorial might provide a first applicable and small step towards it.

Putting it all together this tutorial will provide you with a complete set of concepts to deal even with complex software integration issues in your development.

Scope

The targeted audiences are software quality and test managers, project managers, project leaders and senior software architects. Academics or researchers may also find the topic "*Post Integration*" – how to create transparency without heavy weight processes a highly interesting research area. Novice developers might join as well, however it is assumed that they have read the provided case and the papers really carefully in order to gain a deep insight into serious software system integration problems. Furthermore the participants should have had at least some basic experience with a version control system and therefore as well experience with software development.

References

[1] Thomas Zimmermann, Andreas Zeller, E. James Whitehead, Jr., Sunghun Kim; “Predicting Faults from Cached History”, ICSE 2007, Minneapolis

[2] Zeller, Andreas, “The Future of Programming Environments: Integration, Synergy, and Assistance”, ICSE 2007, Minneapolis

Author: Jürgen Salecker, Siemens AG, CT SE 2

e-mail: juergen.salecker@siemens.com