knowit

Integration Driven Development Large Scale Software Intensive Systems

Pär Hammarström Senior Mgmt Consultant Knowit Technology Management par.hammarstrom@knowit.se 072 202 6277



knowit

DEVLIN 2013 Linköping

What is the Best Development Method?



Contingency theory is a class of behavioral theory that claims that there is no best way to organize a corporation, to lead a company, or to make decisions. Instead, the optimal course of action is contingent (dependent) upon the internal and external situation.

The Hidden Software Industry







Up to 80 percent of the development cost is software related

Most of the innovation is software related

Scale IS a Problem



What is Large Scale Product Development?

A Shared Understanding about What to Build!



The Forest and the Trees

The Whole >> the Sum of its Parts



Emergence comes from hindsight

A Social Construct



Community of Interest



- Common Point of Reference
- Different Meanings
- Means of Coordination and Alignment
- Means of Translation
- Plastic Working Arrangements
- Different Concerns Simultanously

Boundary objects are objects which are both plastic enough to adapt to local needs and constraints of the several parties employing them, yet robust enough to maintain a common identity across sites. They are weakly structured in common use, and become strongly structured in individual-site use. They may be abstract or concrete. They have different meanings in different social worlds but their structure is common enough to more than one world to make them recognizable, a means of translation. The creation and management of boundary objects is key in developing and maintaining coherence across intersecting social worlds.

A Shared Mental Model

Dialogue through Visualizations

Design as a Duality – Seeing the Forest and the Trees



Technology Management



Different Views



Dependencies Governs Integration, Verification & Validation



The Anatomy day



How to Eat an Elephant?



Integration Plan



All Large Systems Starts as Small Systems



Scaled Agile Framework (Example)



A Lightweight adaption from Dean Leafingwell's Scaled Agile Framework

Summary

- A shared mental model of the system
- Focus on dependencies
- Create pull from validation needs
- Just enough intentional architecture
- Done is the engine of more
- LIGHTWEIGHT ... it depends

knowit