Experience with Software Methods and Global Delivery

Matt Flaherty, Collin Murray,
Asbjørn Rolfing, Stanislav Rost,
Kamran Shahroudi, Augustin Tibazarwa

Summary of Presentation

- Agile development introduction
  - **Collin:** Adoption of agile
  - **Asbjørn:** Innovation and software development
- Between the waterfall and radical agile (XP)
  - **Gus:** Experiences with CMMI
  - **Kamran:** Experiences with MBSD
- The missing term of the SD formula
  - **Stan:** Managing software developers
- Development of open-source projects
  - **Matt:** Developing Eclipse
- Global Delivery

MITSloan MANAGEMENT

Collin Murray

ADOPTION OF AGILE

Waterfall to Lean/Agile Goals

- Faster time to market
  - Last 3 major releases averaged 2.7 years
- Responsive to customer needs as they change (provide value)
- Higher quality software
  - Reduces support/maintenance costs
    - The best code is no code…

MITSloan MANAGEMENT

Transition Problems

- Lack of fully automated testing
  - Difficult to perform quick, daily testing
- Existing technical debt
  - Thrashing between new features and bugs
  - Code size greatly increased
- Culture of waterfall projects
  - Need champions/experience with agile
  - Rewards system is individual based
  - Some teams still use waterfall

MITSloan MANAGEMENT

Factors to Ease Transition

- Technology leads very strong, recognized the advantages of agile
- Executive level support
- Frequent (if not daily) builds
- Large test base
  - Customers (strong relations), internal deployment (large test bed)
- Strong source control

Asbjørn Rolfing

INNOVATION

Innovation & Software Development

	Agile	Waterfall
Creative & failure-tolerant culture?	?	√-
Focus on high value creation?	?	√
High maneuverability	?	√-
High speed	?	√+
Innovative	**?**	**√/√-**

MITSloan MANAGEMENT

Agile supports innovation

	Agile	Waterfall
Creative & failure-tolerant culture?	√+	√-
Focus on high value creation?	√+	√
High maneuverability	√+	√-
High speed	√	√+
Innovative	**√+**	**√/√-**

MITSloan MANAGEMENT

Stanislav Rost

SOFTWARE IS PEOPLE

Managing Software Engineering

- People + Process = Program
  - Shouldn't the "process" also focus on the "people?"

- In practice, people/team management is key
  - Forming complementary, self-propelling teams
    - Enough to accomplish a clear intermediate goal
  - Assigning the right tasks to the right people
  - Designing incentives and motivations
  - Conflict avoidance and control

MITSloan MANAGEMENT

Software Developers

- Fresh out-of-college recruits…
  - Slept through the Software Engineering class
  - Used to small projects, SE perceived as overhead
  - …but, possess and follow a basic "natural" SE cycle
- How do engineers feel about real-world SD?

The Good	The Bad	The Ugly
Similar to "natural" SE Concrete tasks to work on Division of labor Organizational structure Growing and learning	Documentation Dependencies on other people Technical disagreements Culture clash	Goals keep changing Territory and credit disputes The caged-bird feeling The unused potential

MITSloan MANAGEMENT

Software Developers: Lessons

- Productivity > intelligence
  - Best predictor of productivity:
    quantity x quality of completed projects
- 80% of work is boring, but assign no boring work
  - Highlight/invent a challenge in any task
  - Make a boring deliverable a waypoint to a better one
- Offer ownership and give proper credit
  - The only forms of payment are money and self-esteem
- Provision for developers "moving on"
  - Developers should co-write and cross-review code

MITSloan MANAGEMENT

Augustine Tibazarwa

CMMI

CMMI Framework
Why bother ?

- Needed framework for (software) engineering to
  - Improve time-to-market
  - Lower Development cost
  - Discriminate among offshore services
- Prior effort
  - CMM …software only … stages
  - ISO 900x focused on documents

Gus Tibazarwa, *atibazar@mit.edu*

CMMI Framework

Link to Business Strategy ?

-Multiple products & platforms

-Market opportunity

-Global/Distributed Delivery

CMMI Level 3 practices	Practice Areas
Process	Organizational Process Focus
	Organizational Process Definition
	Organizational Training
Support	Configuration Management
	Process and Product Quality Assurance
	Measurement and Analysis
	Decision Analysis and Resolution
	Organizational Environment for Integration
Project Mgmt	Project Planning
	Project Monitoring and Control
	Supplier Agreement Management
	Integrated Project Management
	Risk Management
	Integrated Teaming
Engineering	Requirements Management
	Requirements Development
	Technical Solution
	Product Integration
	Verification
	Validation

CMMI Framework
Verdict ?

+Standard Framework: *Communication, Comparisons*

+Available Training

- Requires continual sell

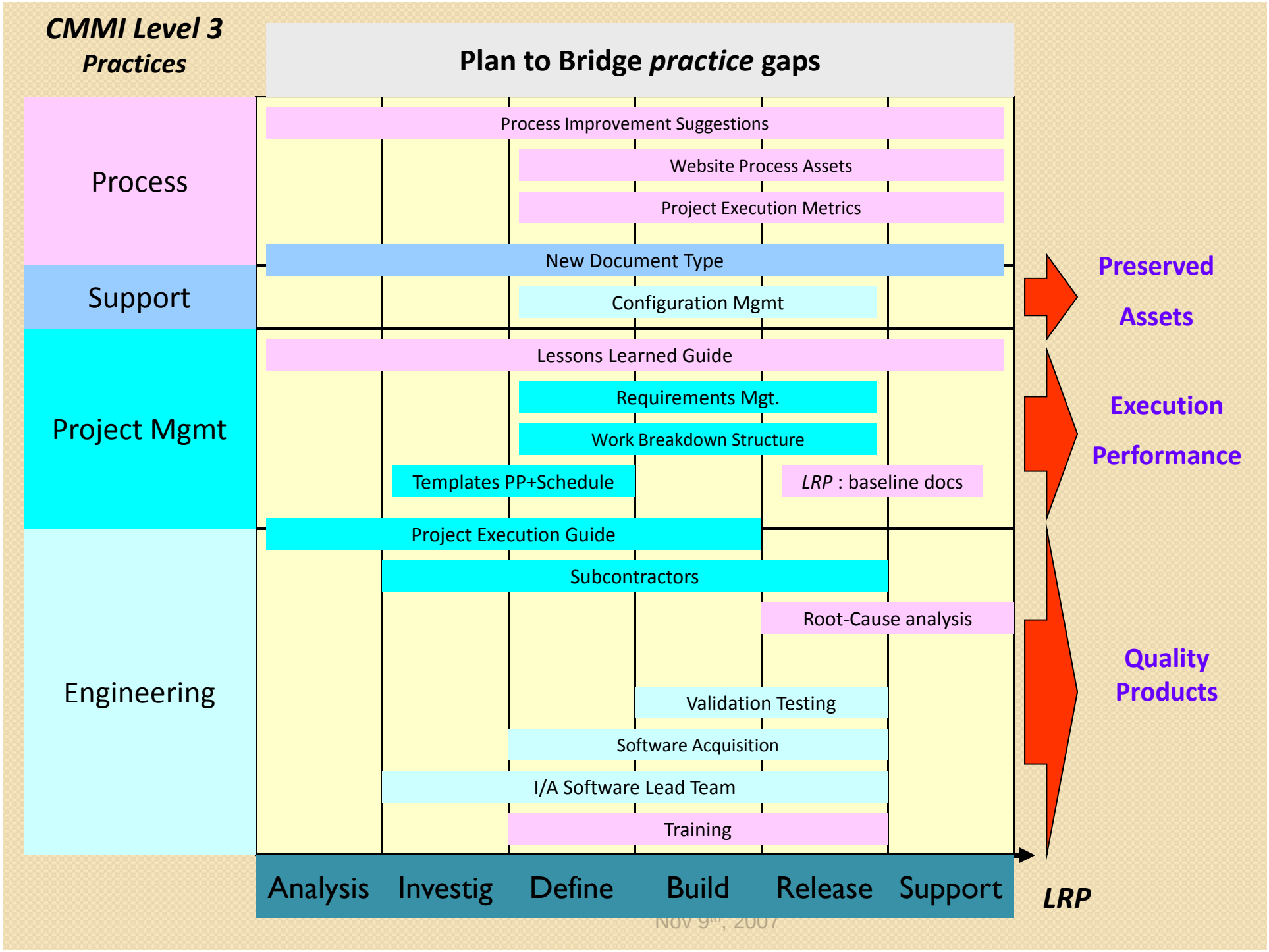
- Requires organizational overhead

- Change → Results time-lag

- Empowers organization … not individuals

MITSloan MANAGEMENT

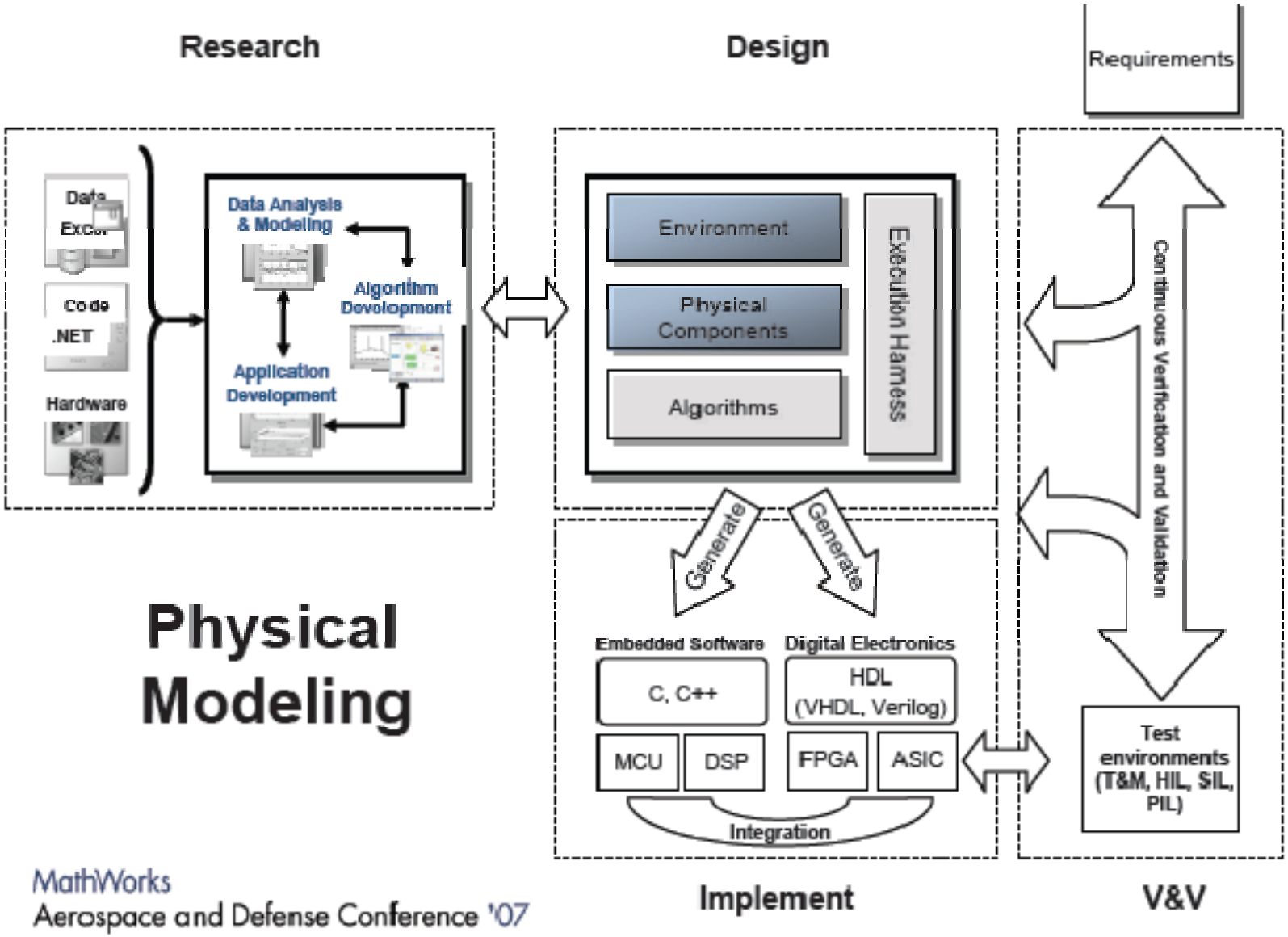
Gus Tibazarwa, *atibazar@mit.edu*


CMMI Level 3
Practices
Plan to Bridge practice gaps
Process
Support
Project Mgmt
Engineering
Process Improvement Suggestions
Website Process Assets
Project Execution Metrics
New Document Type
Configuration Mgmt
Lessons Learned Guide
Requirements Mgt.
Work Breakdown Structure
Templates PP+Schedule
LRP : baseline docs
Project Execution Guide
Subcontractors
Root-Cause analysis
Validation Testing
Software Acquisition
I/A Software Lead Team
Training
Preserved Assets
Execution Performance
Quality Products
Analysis
Investig
Define
Build
Release
Support
LRP

Kamran Shahroudi

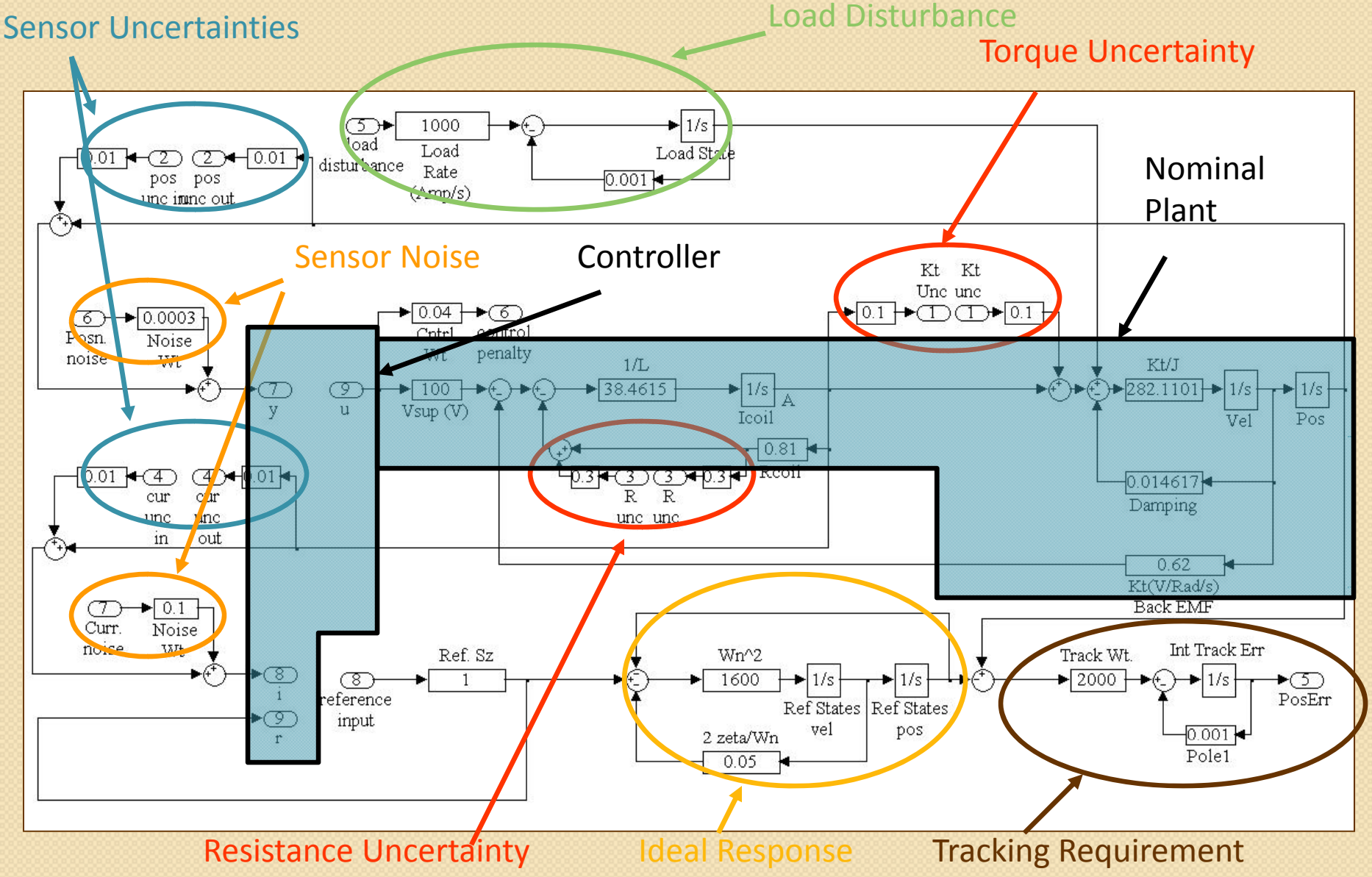
MODEL BASED SOFTWARE DEVELOPMENT AND SYSTEM ENGINEERING

What is MBSD (MBSE) ?



MITSloan MANAGEMENT

What does a Model look like?



Where is it being applied successfully ?

- Safety Critical/ Embedded Code Applications:
  - Turbine Fuel System Controls (My own experience), F35 JSF (Lockheed), Satellite Communication (BAE) and Control (Swedish Space Systems), Automotive Engine Control (Almost all OEM's)

- High Complexity mixed with High Robustness and Performance Requirements.
- High Levels of Uncertainty Regarding the Physical Processes and States of the System.
- Mechanical Systems -> electronic systems -> communication buses and distributed embedded controllers -> More Software!.

MITSloan MANAGEMENT

My own experience with MDSD, MBSE

- Very Fast Development Cycles
  - Only needed 1 iteration and took a few weeks.
  - Did not have to change the autogenerated code.
- Easier to cope with design changes.
- Easier to detect coding problems or design mistakes.
- Enabled more distributed and modular development.
- Not all the benefits were due to MBSD alone:
  - Robust Controls was the higher abstraction built on top of MBSD.

MITSloan MANAGEMENT

Software Business Aspects of MBSD

- MBSD is a platform
  - Mathworks is Fighting National Instruments (Labview) for dominance.
  - Some Key Standards (e.g. SysML) are emerging not controlled by either of these parties.
- Cultural Factors
  - Resistance: What will System Engineers and Software Engineers do?
    - Docs and Code are nowAuto generated.
- Development Process
  - Requirements are now executable (higher abstraction than UML or SysML).
  - Solve Problems at Higher Abstraction and Modular Level.
  - More non software oriented people can now develop and test code.

MITSloan MANAGEMENT

Fit with Iterative and Waterfall

- Fits well with Waterfall
  - Because it helps reduce the number of costly deep (or low level) iterations.
  - I experienced a case where we had only 1 iteration!
- Fits well with Iterative too!
  - Because it enables easy integration of every iteration across teams
  - More Modular development
  - Gives higher level engineers from multiple disciplines or multiple organizations a better platform for collaboration.

MITSloan MANAGEMENT

Matt Flaherty

DEVELOPMENT OF OPEN-SOURCE PROJECTS

Open Source: History & Motivations

- Historically:
  - Solve common broad 'pain points', both technical and financial
  - Best quality when 'scratching own itch', 'eating own dog food'
  - Thus, early solutions often arose in a datacenter/backoffice context
    - Linux OS, Apache Web Server
- Success of business and development model has led to adoption in other contexts:
  - Eclipse Java Development Environment
  - Apache Jakarta libraries for Java
  - Java Programming Language
  - OpenOffice Document Editors
- License and openness of process also drives participation and success
  - Some viral – GPL, Commercially friendly – APL/EPL, Open – BSD/MIT
  - E.g.: Sun's perceived restrictive license on OpenOffice

MITSloan MANAGEMENT

Open Source: Procedure Highlights

- Not a true network organization – successful projects have hierarchy
  - Common terminology involves committers and contributors
  - Committers maintain specific areas of source base
- Small number of developers typically do most of the work
  - Economies of scale from community testing, patch contribution
  - Patches provide a first-stab at a problem solution
- Efficient and classically agile processes
  - Meritocracy built into culture, often very 'alpha' interaction
  - Code reviews enforced by embedded hierarchical structure
  - Daily (sometimes continuous) builds used

MITSloan MANAGEMENT

Personal Experience: 'The Eclipse Way'

- Development is a fixed 1 year cycle
  - Ships every year, regardless of pressures – Iron Triangle fixed
  - Classically agile planning: themes -> plan -> iterations
- 6 week milestones
  - Plan is reviewed and updated with each milestone
  - Milestones must be ship-quality, and are used by real users
- Other key takeaways
  - Feedback is key to success – "Eating own dog food"
  - Modular architecture is a process enabler
  - Transparent process lessens communication pain
    - Overall plan and subteam plans are public
    - Eclipse foundation enables communication with technology
  - Every build must be useful to someone
    - Nightly build consumable by Eclipse teams
    - Milestone builds consumable by the community

MITSloan MANAGEMENT

Collin & Stan

Global Delivery Methods

Global Delivery
U.S., Japan, India, Israel, China

Pros

- 24x7 coverage
- Available skillset
- Cheaper cost for some locations
  - At least initially
- Knowledge of local customs & language
  - Dealing with local customers

Cons

- No colocation
- Cultural differences
  - Holidays
    - e.g.: Golden Week
- Time zone differences
- Language barriers
- Export restrictions
  - e.g.: security code
- Lack of authority
  - In country manager

MITSloan MANAGEMENT

Global Delivery

- Working with globally distributed teams
  - Communication cycles are longer
    - The "24 hour penalty"
  - Documentation/goals/expectations are 10x more critical
    - Misunderstandings can lead to humongous bugs
  - On-site team manager is a must
    - Can't coordinate what you can't see
    - Must know the whole picture, can answer questions quickly
  - Mind the cultural differences
    - Some people won't say "no", you have to infer it

MITSloan MANAGEMENT

BONUS SLIDES

Concluding Slide

- It's hard to coordinate six busy people to produce a few slides
- Imagine coordinating hundreds of software developers!

MITSloan MANAGEMENT

Innovation & SW Development

	Agile	Lean	Waterfall
Creative & failure-tolerant culture?	XP, achieve an early win and rapid feedback	Eliminate waste but amplify learning	Do it right the first time
Focus on high value creation?	Close customer partnerships	Iterative approach , value stream mapping	Specify details clearly
High maneuverability	High change tolerance	Conceptual integrity	Water only falls one way
High speed	Quality in design	Just-in-Time	But it gets to the ground faster
Innovative	√+	√	√/√-

MITSloan MANAGEMENT

The Design Process