Implementing CMMI-based Process Improvement Using the Rational Unified Process

John B. Miller – SEI Authorized CMMI Lead Appraiser
Rolf W. Reitzig – SEI Resident Affiliate

Tuesday, November 14th, 2006

cognence inc
Improving Software Economics

First Off...

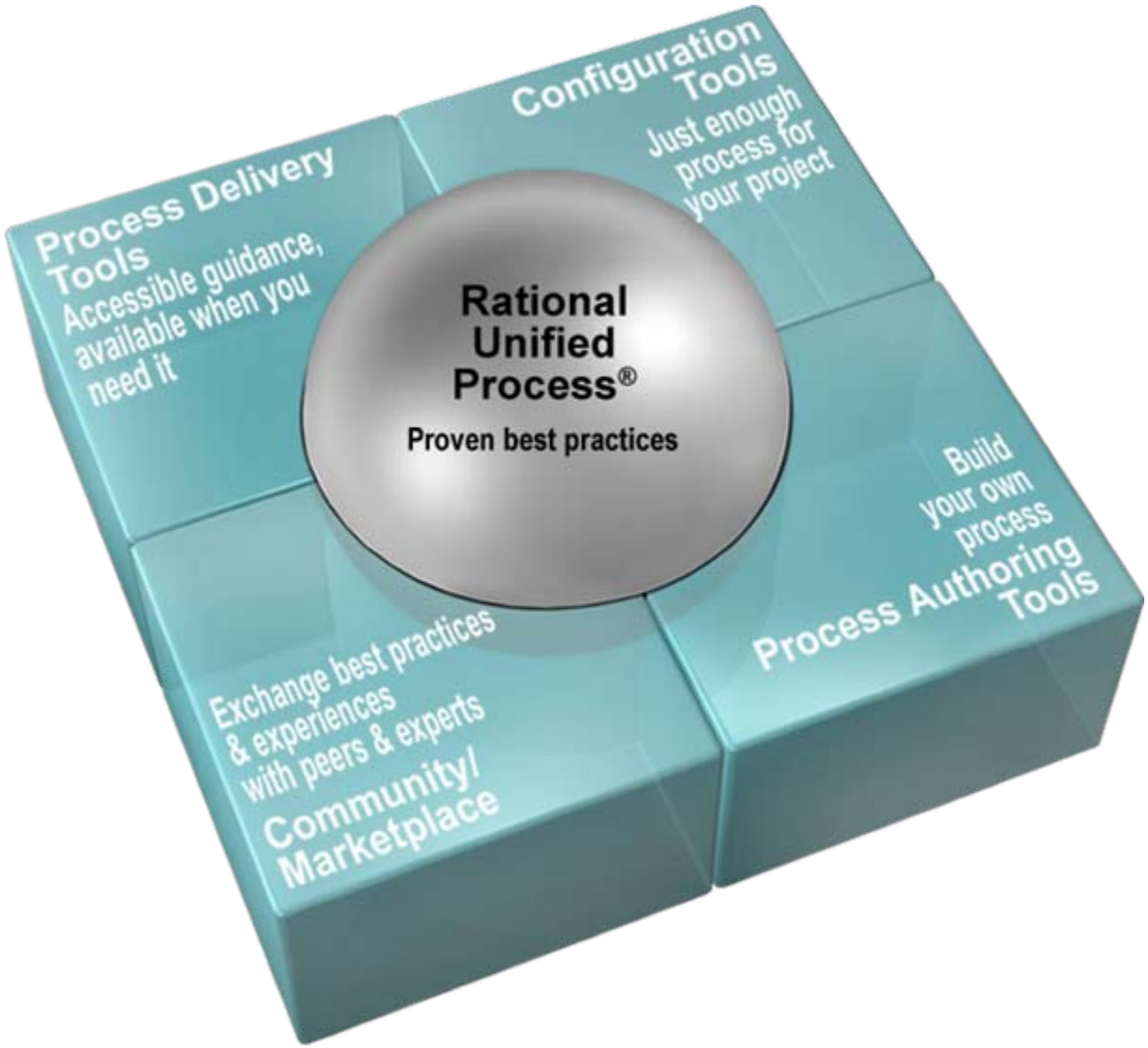
Why are you here?

Rational Unified Process Overview

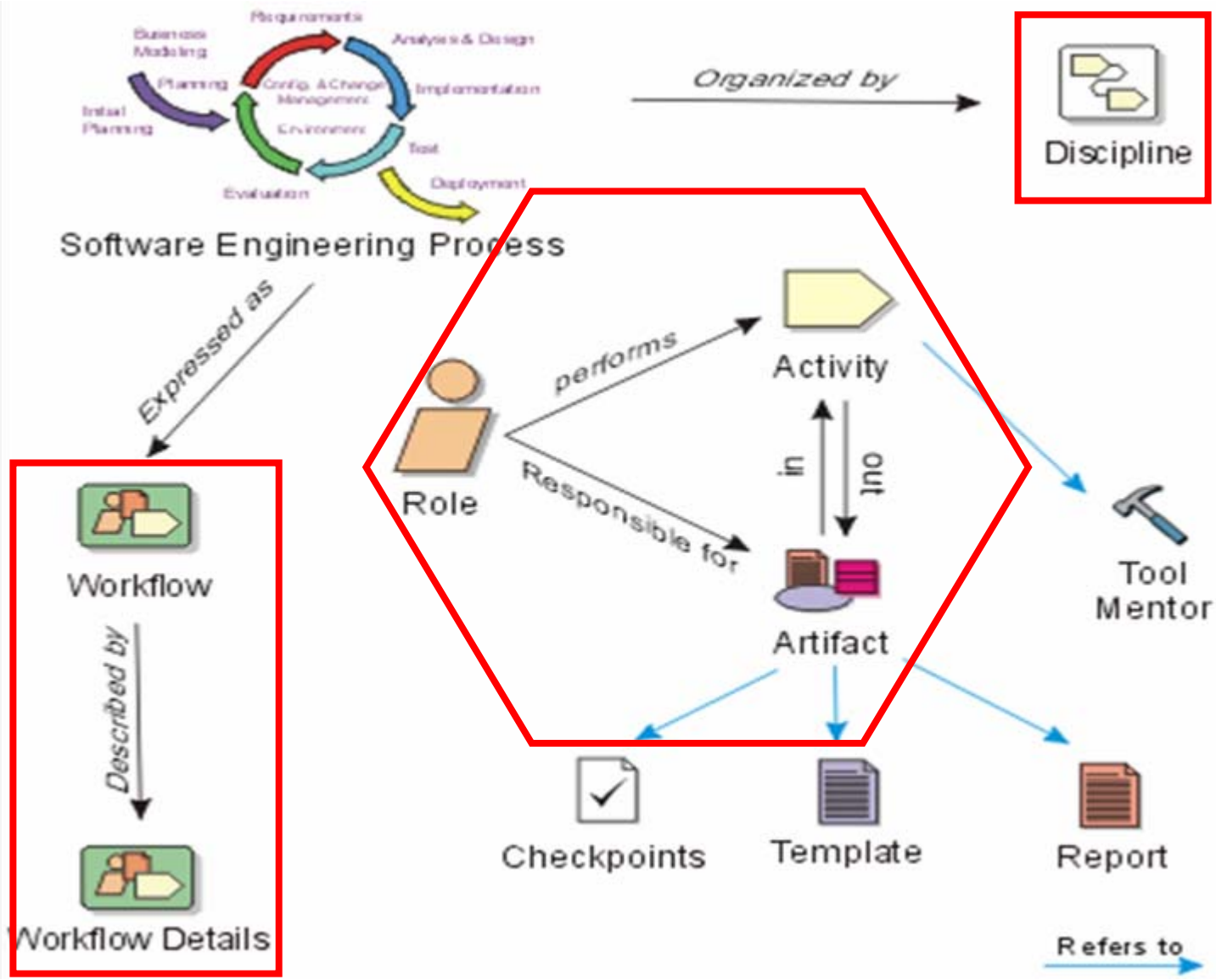
cognence inc

Improving Software Economics

Introducing the RUP Platform

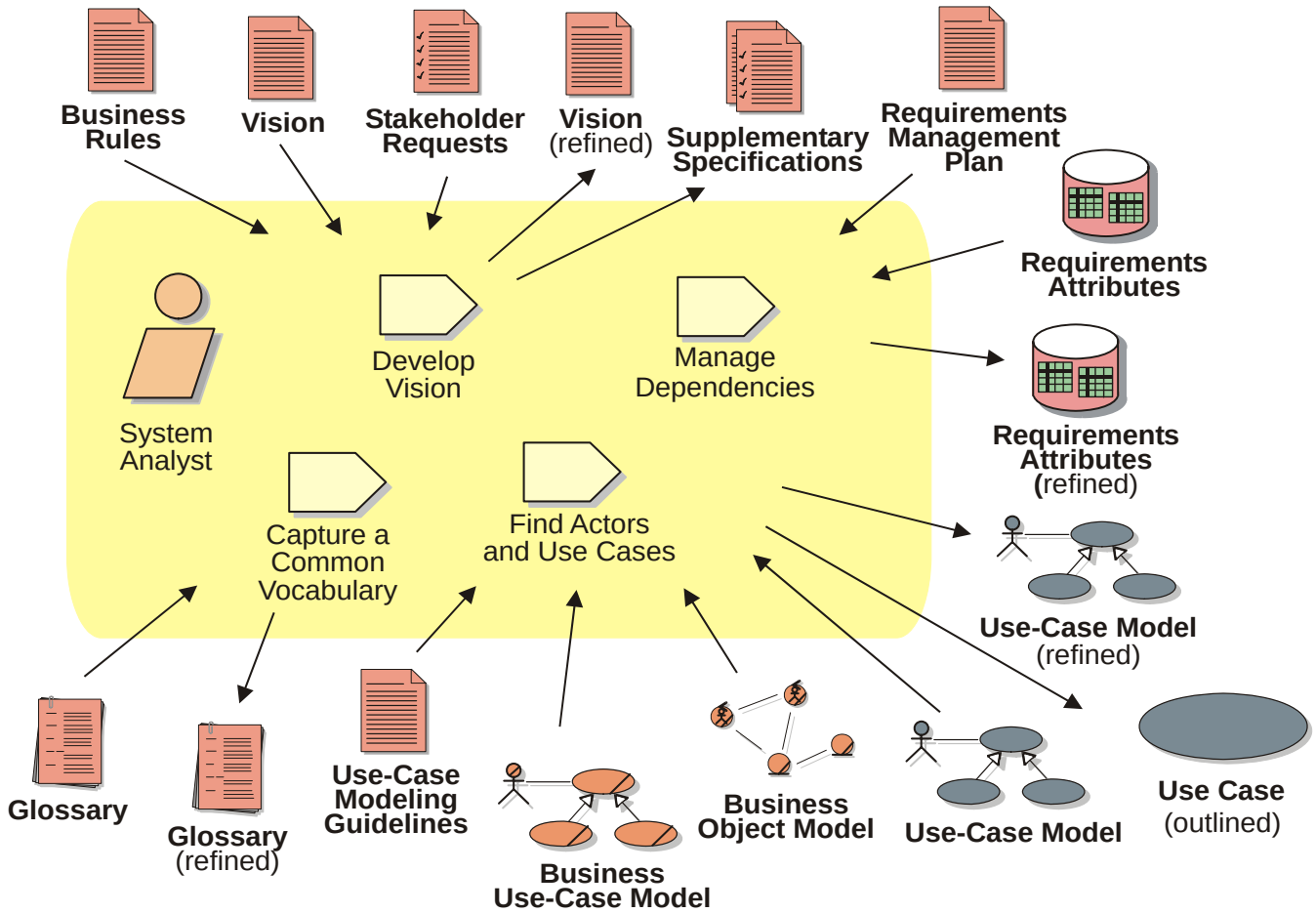


Basic Structure of RUP: Key Concepts



Roles Perform Activities and Produce Artifacts

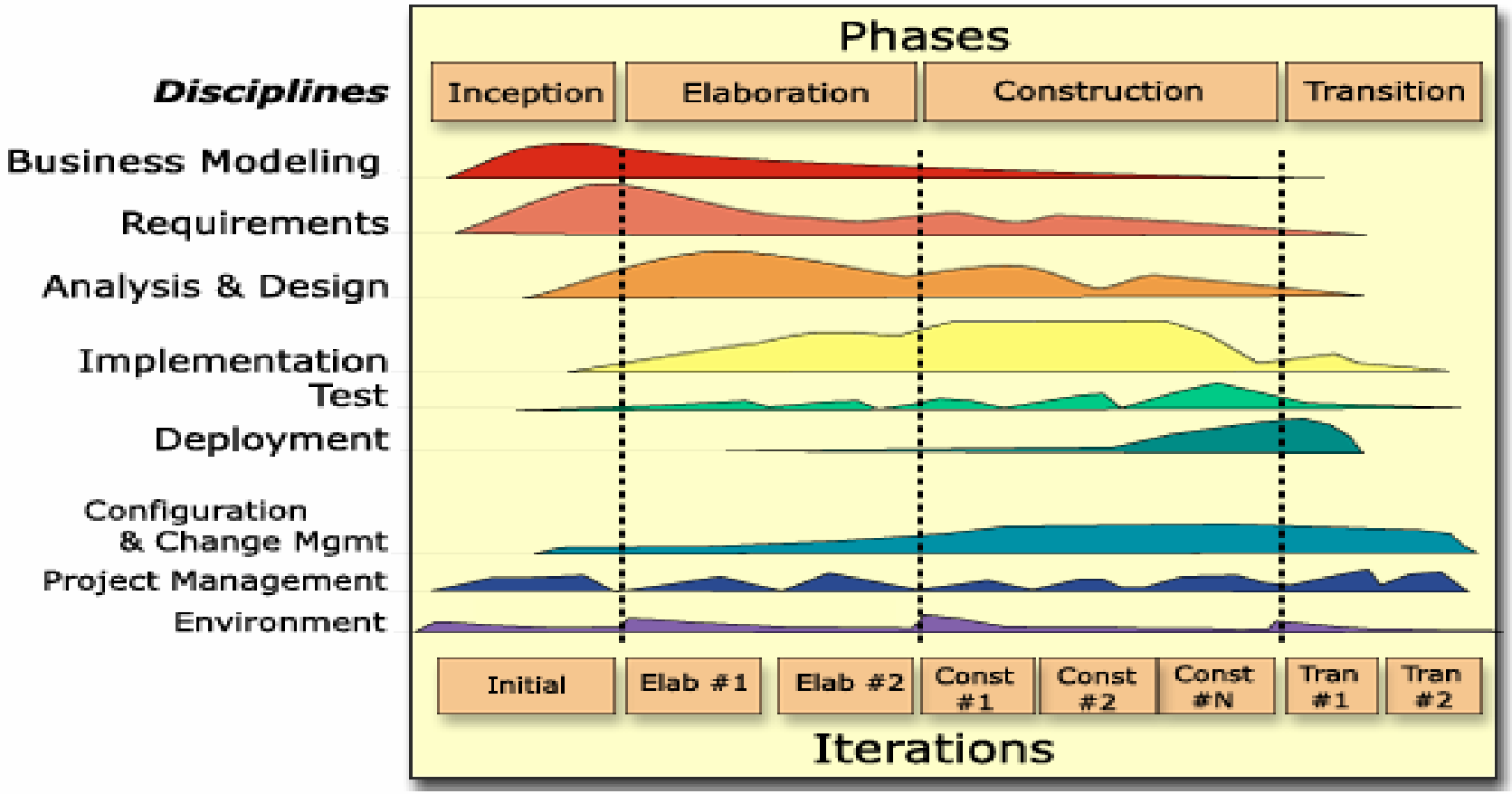
Example:
Requirements->
Workflow Detail->
Define the System



The Spirit of RUP

1. Attack major risks early and continuously… or they attack you
2. Ensure that you deliver value to your customer
3. Have a maniacal focus on working software
4. Accommodate change early in the project
5. Baseline an executable architecture early on
6. Build your system with components
7. Work closely together as one team
8. Make quality a way of life, not an afterthought

Iterative Development



What Will RUP Get You?

- A collection of world best practices
- A mechanism for modifying them to suite your needs, as well as integrating other best practices and your own practices
- A mechanism for tailoring and deploying them to projects
- With Rational Portfolio Manager, a mechanism for managing projects according to projects' defined processes, and gathering important data critical to data-driven project management & future planning

What Will RUP Not Get You?

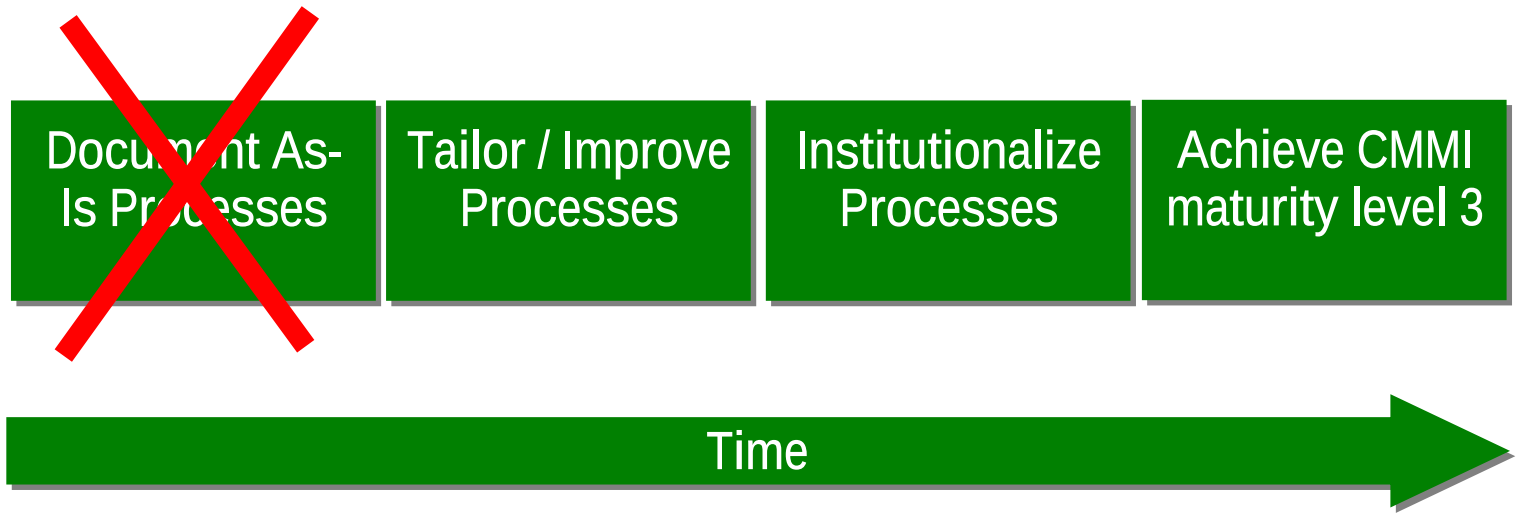
- A guarantee that you'll perform better without some effort on your part
- Instant expertise in software engineering best practices
- Compliance with all CMMI maturity level 2 & 3 requirements out-of-the-box

How Does RUP Fit In?

- The key to realizing benefits is
  - implementing organizational standard software engineering processes
  - that are deployed (franchised) into projects based on key characteristics
  - and are tailored, managed & measured
- A great way to start is a combination of
  - Rational Unified Process (RUP)
  - Rational Method Composer (RMC)
  - Rational Portfolio Manager (RPM)

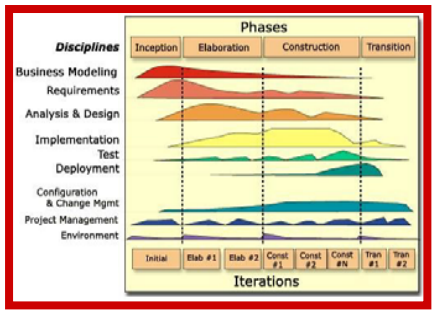
CMMI Maturity Level 3 Acceleration

- Use of the Rational Unified Process and IBM Rational automation tools will accelerate the achievement of CMMI maturity level 3

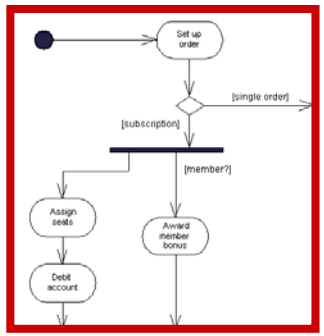


Delivering a Franchisable Process to Your Company

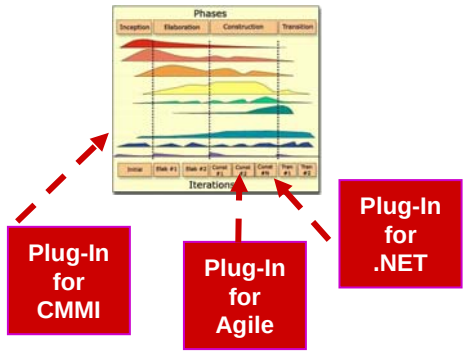
Core RUP



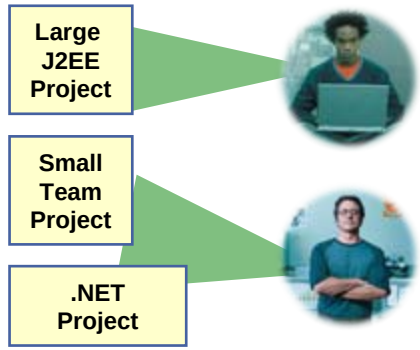
RMC



RMC



RPM



Development organization → **Process engineers, program/project offices** → **Project managers & team leads** → **Practitioners**

Common methodology

- Shared understanding of terminology, deliverables, and responsibilities

Process authoring

- Leverage internal knowledge and process assets

Process configuration

- Configure and deploy process for specific tools, technologies, domains

Process delivery

- Manage project instances
- Gather and report metrics

RUP & CMMI

cognence inc
Improving Software Economics

Situation Description

- Evaluated RUP for Large Projects v7.0.1 against CMMI-DEV w/IPPD v1.2 maturity level 3
- RUP's processes, tasks, templates, etc. were reviewed and evaluated for compliance
  - Determined the risk of meeting the CMMI's intent in each ML 2 & ML 3 practice if the process & associated templates are implemented as described
  - Use of Rational tools was not considered
- Caveat: Actual results might vary based on project implementation and degree of RUP tailoring!

Project Management Process Areas

Strengths

- RUP fundamentally provides a solid project management foundation
  - Planning, Tracking & Oversight, Risk Management

Areas for Improvement

- Supplier Agreement Management – RUP does not address SAM or the notion of suppliers at all
- Estimation and tracking of estimates leaves too much discretion to the project
- Planning for and managing project data – too code-centric

Engineering Process Areas

Strengths

- Overall, RUP's Requirements, Design & Analysis, Implementation, Test, and Deployment disciplines handle CMMI requirements very well

Areas for Improvement

- Traceability without tool support may not be sufficient
- Determining criteria for alternative technical solutions
- Performing build/buy/reuse analysis
- Integration sequence, procedures, and criteria
- Reviews of interfaces for coverage & completeness
- Analysis of peer review data

Support Process Areas

Strengths

- RUP's Configuration & Change Management discipline
- Measurement & Analysis

Areas for Improvement

- Decision Analysis & Resolution – RUP does not provide a process for DAR
- Process Quality Assurance is not called out strongly enough – projects may overlook this important process area
- Identification of configuration items is too focused on engineering work products and doesn't address other work products

Process Management Process Areas

Strengths

- RUP's collection of best practices and mechanism (Rational Method Composer) for tailoring and deploying to a project

Areas for Improvement

- All Process Management guidance is project-centric - RUP does not address organizational-level processes for
  - Organizational Process Focus
  - Organizational Process Definition
  - Organizational Training

Generic Practices

Strengths

- RUP's collection of best practices and mechanism (Rational Method Composer) for tailoring and deploying to a project

Areas for Improvement

- RUP does not provide any policies or guidance for policy development
- Identification, planning, scheduling, and involvement of stakeholders
- Identification and delivery of personnel training
- Feedback loop of project-level measurements and lessons learned to improve the organization's processes and planning capabilities

Sounds Good, But How Do We Get There???

Implementation of software engineering best practices and integrated automation tools across an organization is difficult

Barriers

- Lack of understanding
- Poor planning
- Resistance to change

> "Radical, rapid change is crucial to success; incremental, unfocused changes have a better than 50% chance of failure"
>
> – Gartner

RUP Implementation Roadmap

cognence inc

Improving Software Economics

Roadmap – Setting the Stage

1. Establish Executive Sponsorship with the expectation it is active, not passive
2. Clearly tie the effort to business goals
3. Establish a guiding coalition (MSG/EPG/SEPA) of movers and shakers from across the organization to drive the RUP implementation strategy, approach, and plan
4. Projectize the effort, assign a cost center, and treat it like a project with clear milestones and reviews
5. Tie improvement objectives to each individual's performance review

Roadmap – Tailoring RUP - 1

1. Using Rational Method Composer, develop organizational processes for:
   - DAR, OPD, OPF, OT, SAM
2. Create detailed tasks for:
   - Policy development
   - Estimation & tracking of actuals
   - Identification and delivery of project-level training
   - Objective evaluation of processes
   - Feedback of measurement and lessons learned to organizational process assets
   - Build/Buy/Reuse decision making

Roadmap – Tailoring RUP - 2

3. Improve existing guidance for:
   - Management of non-engineering CIs/project data
   - Stakeholder involvement
   - Traceability
   - Alterative technical solution criteria & evaluation (DAR)
   - Evaluation of peer review data
   - Integration sequence, procedures, & criteria
   - Interface reviews
4. Using *critical thinking,* determine the various RUP configurations your organization might use and establish them as approved lifecycles in RMC

Roadmap – Introducing Improvements

1. Determine gaps in the knowledge and skills required in the organization, train people
2. Pilot on carefully selected projects, provide hypersupport
3. Identify pilot project lessons learned and incorporate improvements into the organization's processes
4. Iteratively deploy more broadly
5. Maintain the feedback loop, continuously improve

End Result

- The outcome will be an integrated, organizationally cooperative process infrastructure (OSSP) that:
  - is the foundation for a successful organizational transformation and improvement
  - provides common processes for measurements & analysis
  - adopts industry best practices as embodied in RUP
  - facilitates software improvement based on consensus priorities
  - Delivers superior engineering results

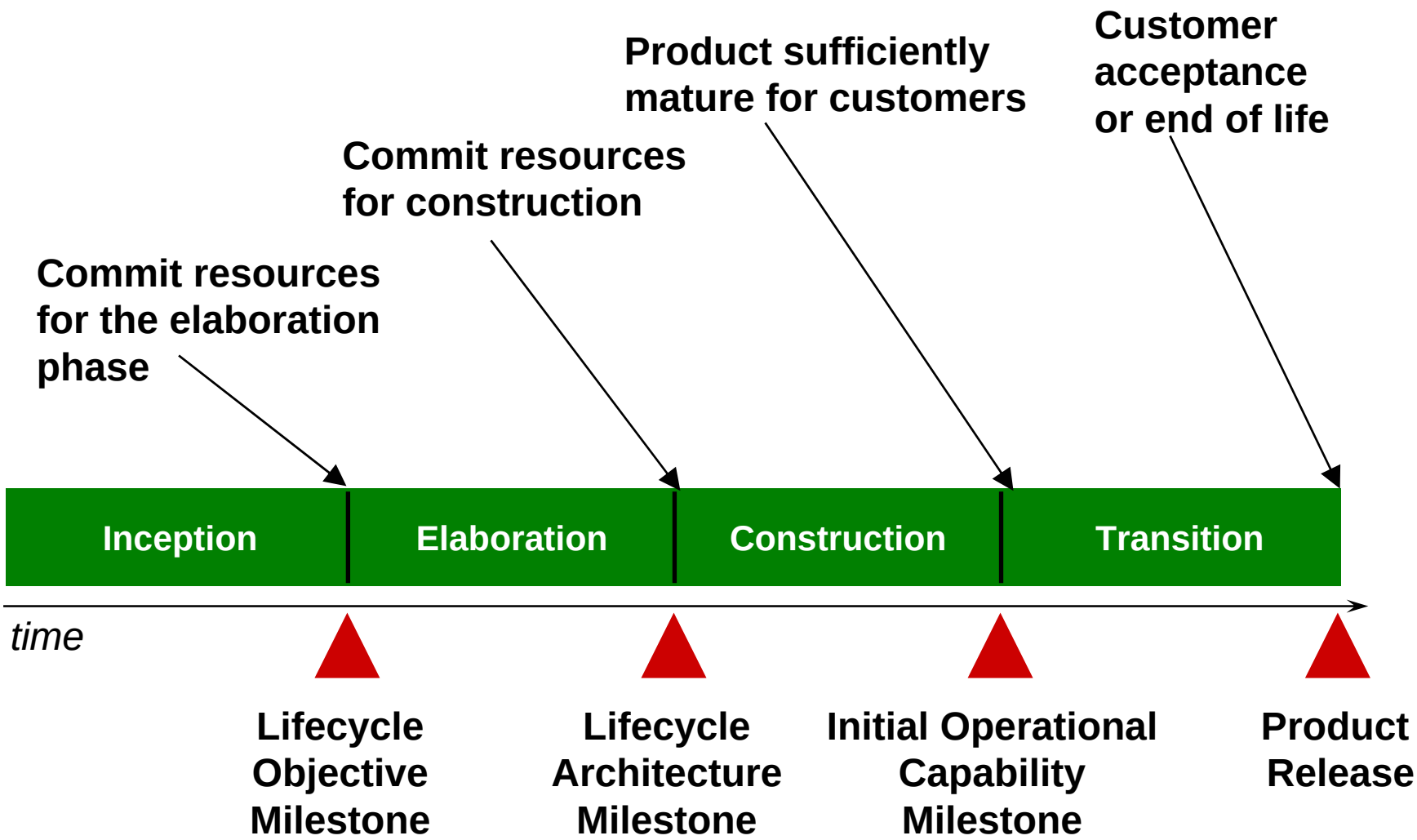
Questions/Discussion

cognence inc
Improving Software Economics

Backup

cognence inc
Improving Software Economics

Major Milestones: Business Decision Points



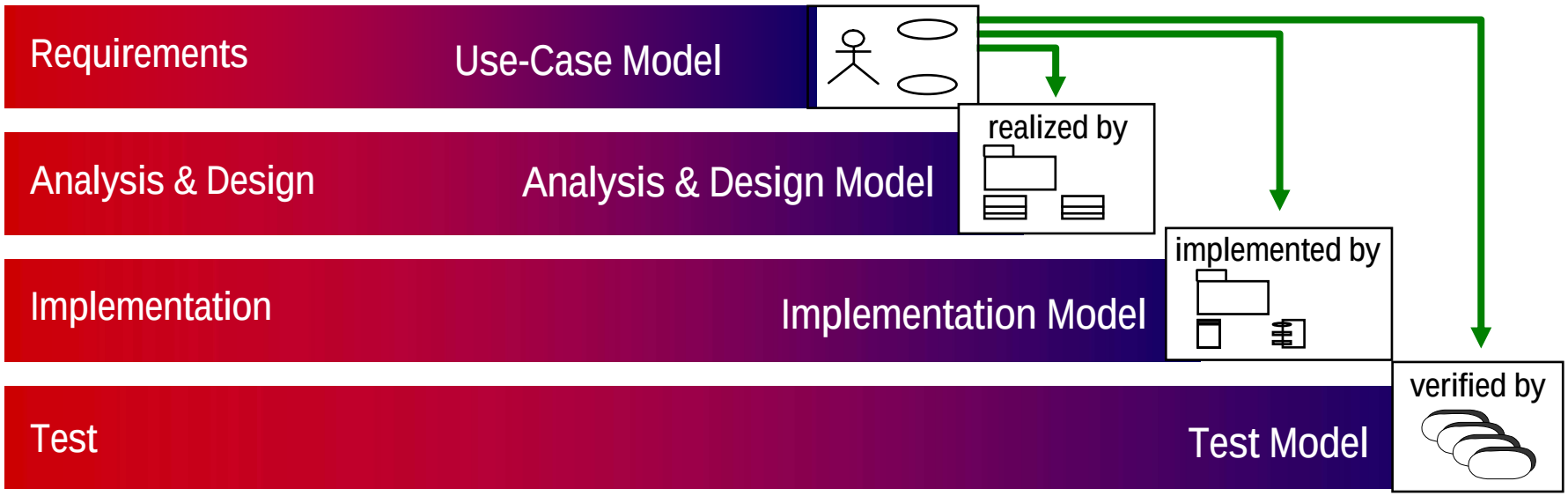
2. Ensure That You Deliver Value to Your Customer

- Focus on key requirements
  - Capture, document
  - Organize, prioritize
- Requirements will change
  - Evaluate impact of change and decide what changes to implement
  - Propagate changes to all team members
- Make requirements accessible

Requirements management leverages your ability to deliver products that meet user needs

Use-Case Driven Development

- A use case describes complete and meaningful services that your system offers to users and other systems
- Use cases drive the work through each iteration
  - Planning of iterations
  - Creation and validation of the architecture
  - Definition of test cases and procedures
  - Design of user interfaces and creation of user documentation

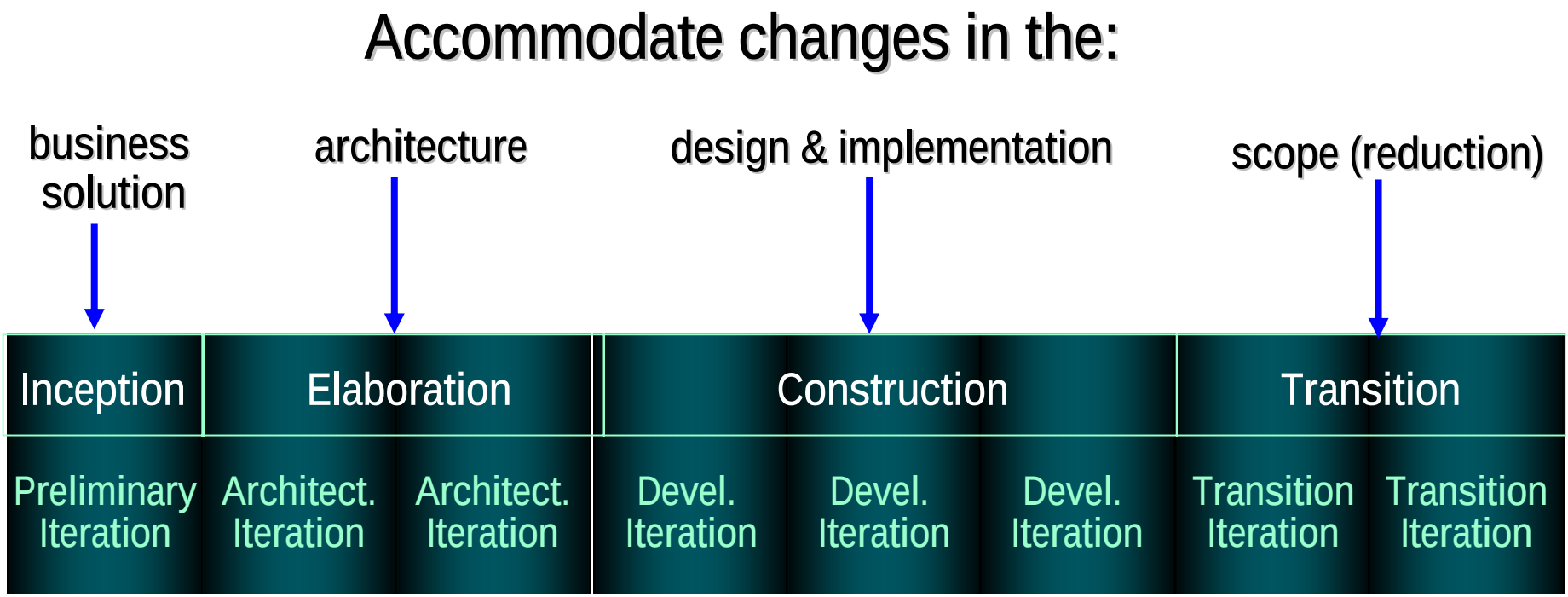


3. Have a Maniacal Focus on Working Software

- Measure progress primarily by reviewing executable code, and test results
  - Plans, requirements, designs and other by-products often provide a false perception of progress and status
- Focus on the final, delivered product, and only the artifacts that matter to get at this goal consistently
  - Streamline the process
  - Do not use all of the RUP! Only use what makes sense to your project – tailor it to suite your organization's and project's needs.

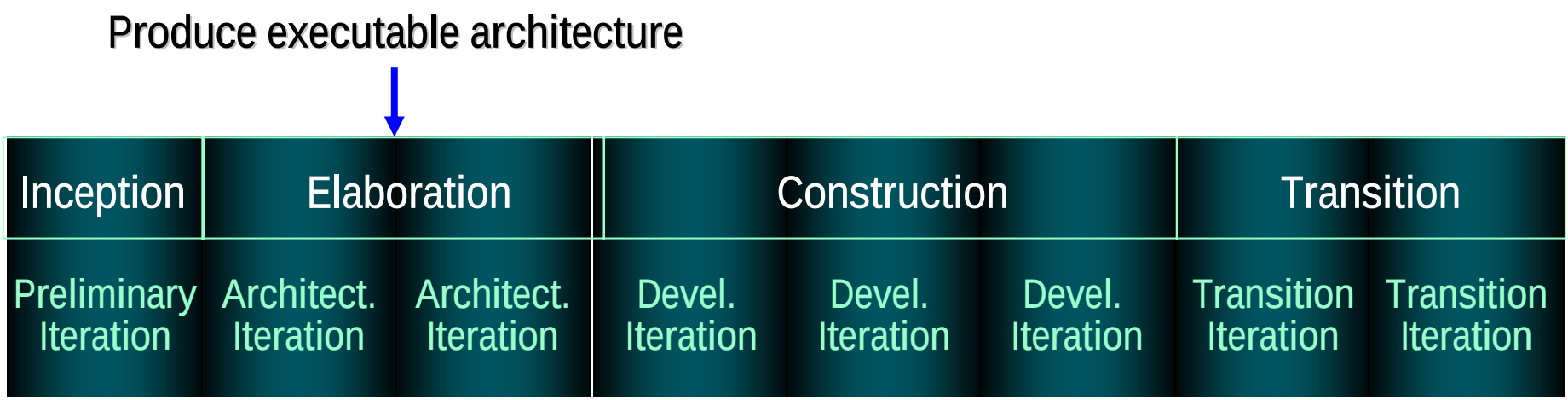
4. Accommodate Change Early In the Project

- Today's systems are too complex to get the requirements, architecture, design, implementation and scope right the first time



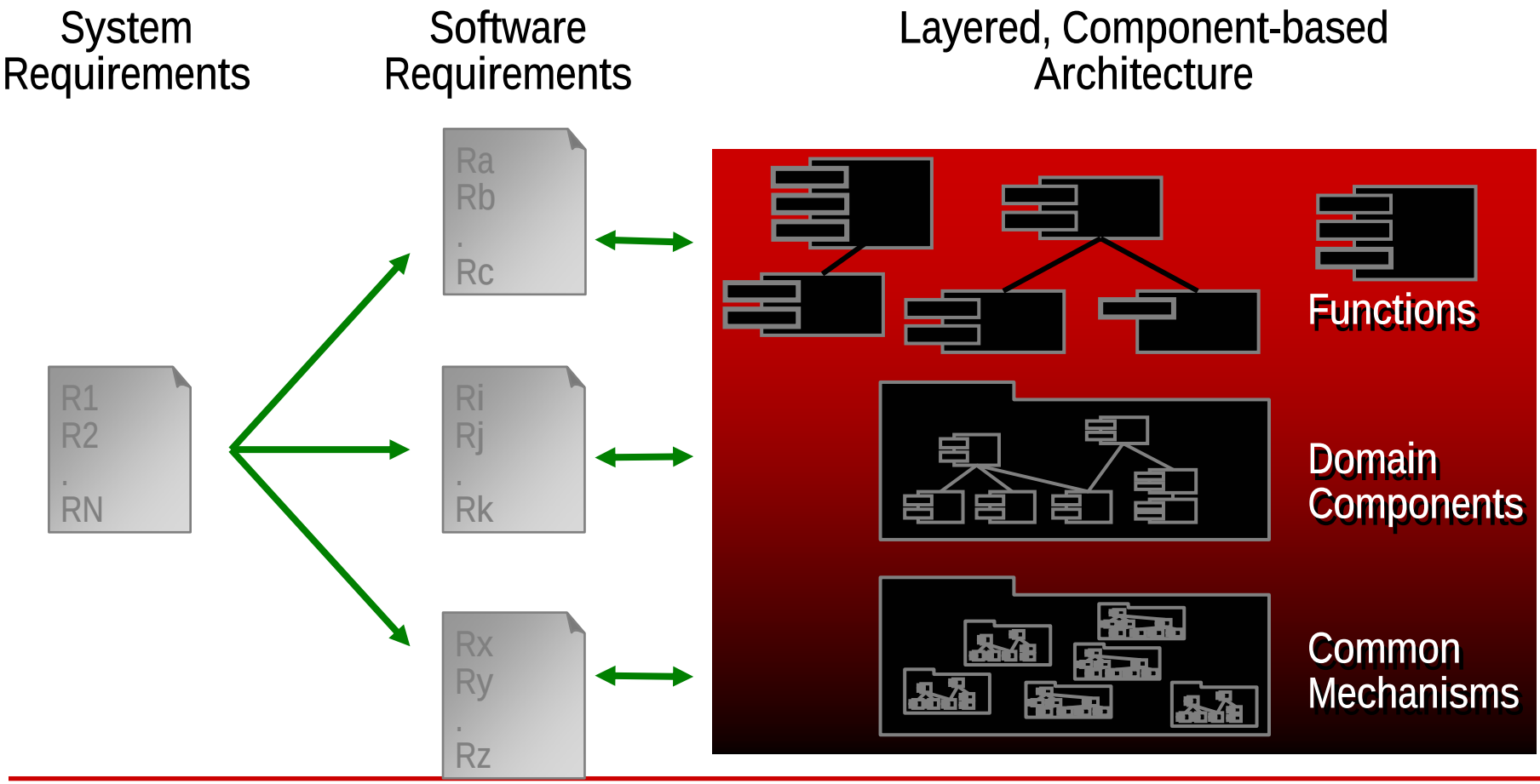
5. Baseline an executable architecture early on

- Architecture provides a skeleton structure of your system
  - Subsystems, key components, interfaces, architectural mechanisms (solutions for common problems, such as persistency, inter-process communication, ..)
- Implementing and testing the architecture mitigates most technical risks



6. Build Your System With Components

- Architecture-driven, Component-based Design
- Component architecture provides flexibility



7. Work Closely Together As One Team

- Empowered and self-managed
  - Clear vision
- Accountable for team results
  - Clear expectations
  - All for one, one for all
    - avoid “My design was good, your code didn’t work”
- Optimized communication
  - Effective process (right-sized for your project)
  - Organize around architecture, not around functions
  - Get the right tool support
    - Easy access to current requirement
    - Private workspaces
    - Easy access to defects….
    - …

8. Make Quality a Way of Life, Not an Afterthought

Software problems are
100 to 1000 times more costly
to find and repair after deployment