



INT Agile Process

Steven Reynolds
INT

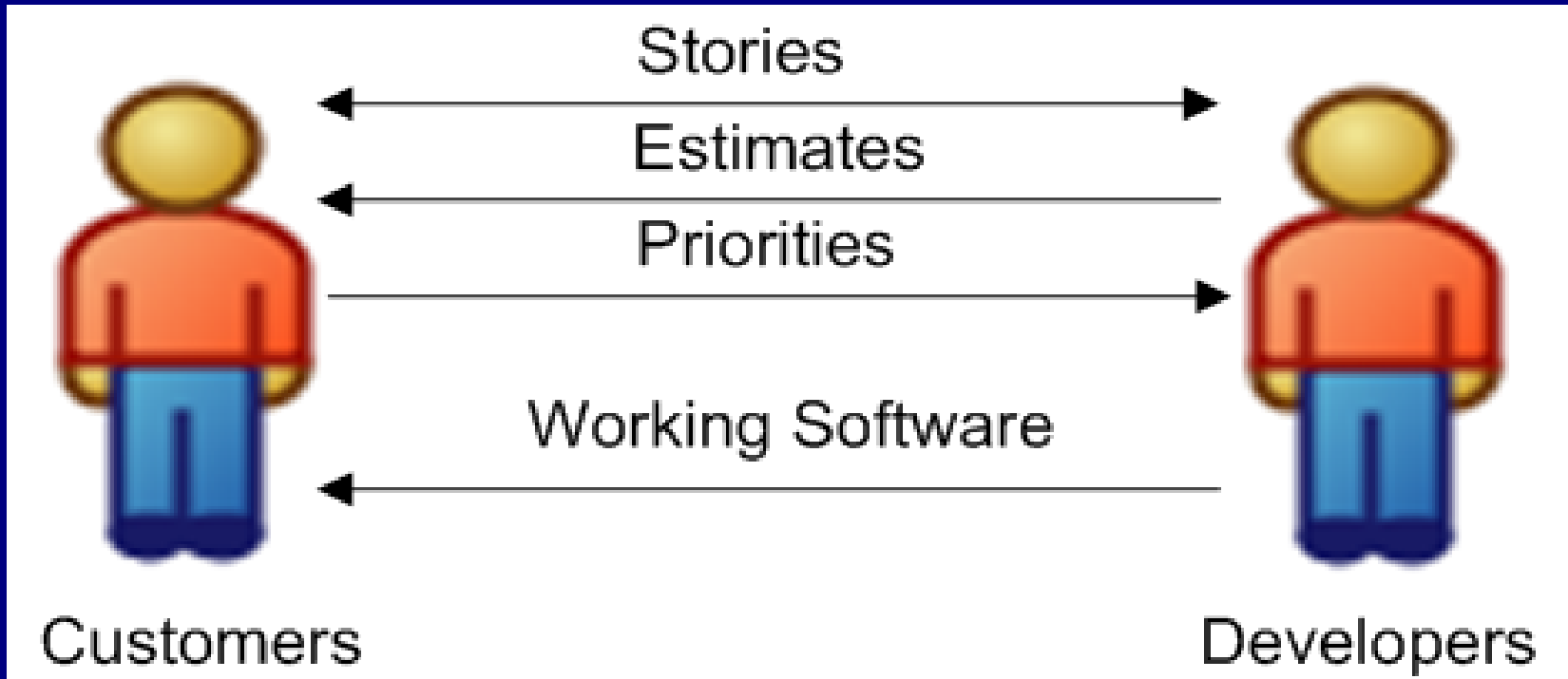
SEG

September, 2007

Agile Benefits

- Builds the right software – constant feedback ensures it is right
- Allows changes without costly rewrites
- Provides continuously updated estimates for remaining work
- Adds most valuable features first
- Frequent builds reduce risks – assumptions can be tested in the real world

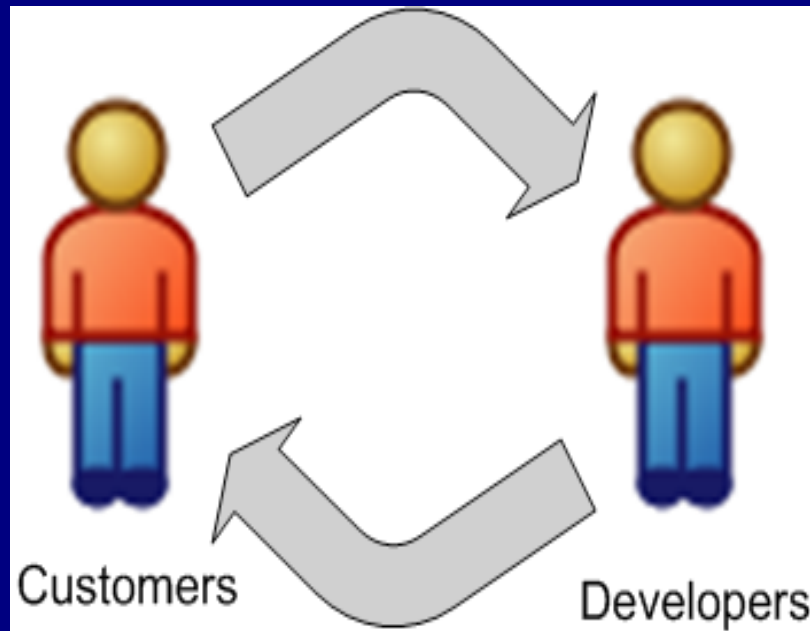
Individuals & Interactions



- A user story is an informal use case
- The process enables rich interactions
- Does *not* require a heavy planning phase up-front
- Does require customer interaction throughout

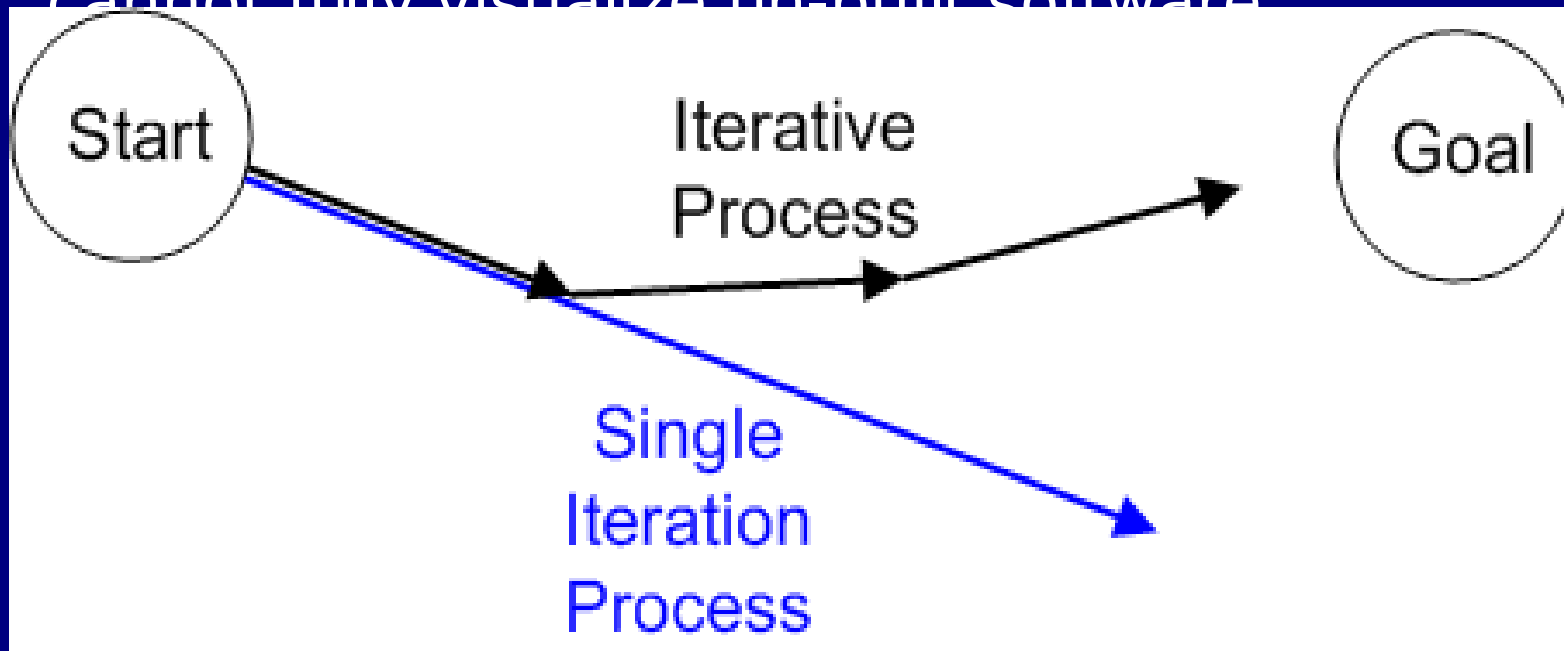
Iterative Development

- Short iterations – two weeks is the norm at INT
- Enables rich interaction – Customer examines the working code
- Allows course correction (adaptive feedback loop)



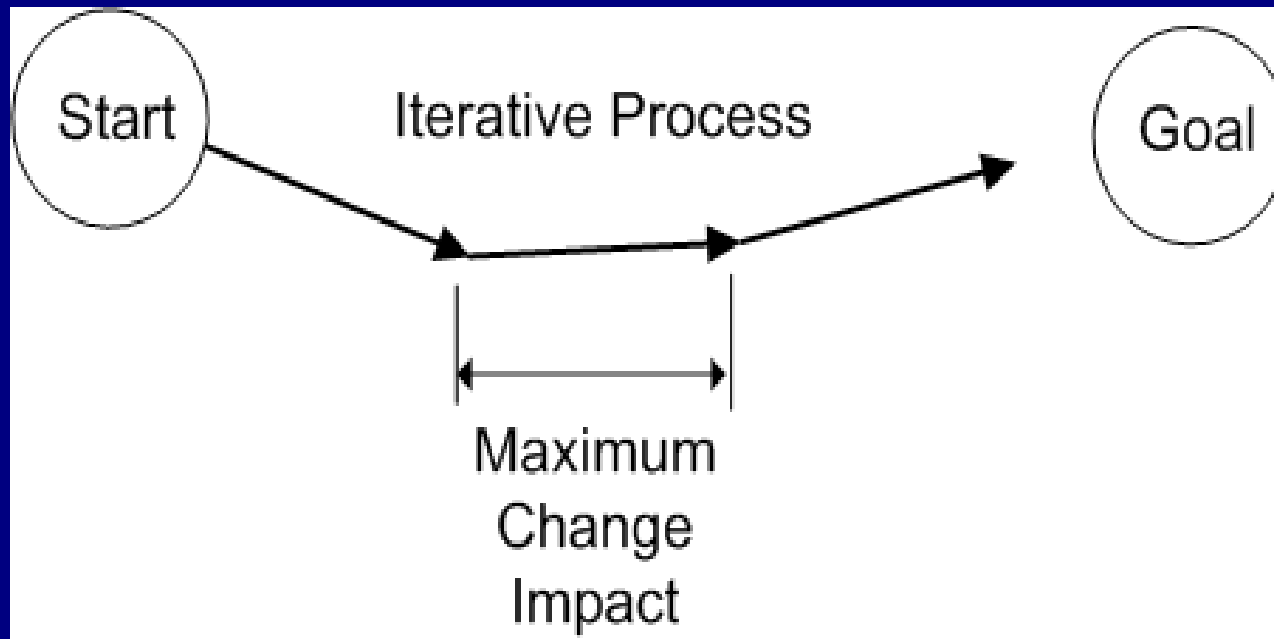
Iterative Development

- Feedback gives better results
 - Gives corrections to our imperfect understanding
 - The goal changes: the market changes and competitors change
 - At the start, we cannot see the goal very well. We cannot fully visualize un-built software



Responding to Change

- Code level planning is limited to scope of current iteration
- Limits impact of change



- Change is expected and A-OK

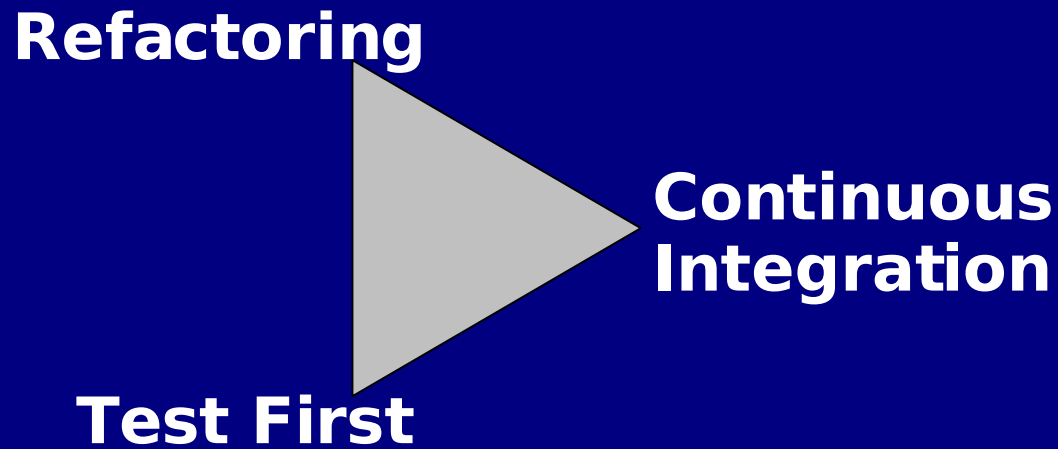
Planning

- Planning happens at many levels
 - Release
 - Iteration
 - Individual story
- Only plan for what you are just about to do
- Pay as you go, *not* all up-front

Agile Values

More Valued	Less Valued
Individuals and Interaction	Process and Tools
Working Software	Comprehensive Documentation
Customer Collaboration	Contract Negotiation
Responding to Change	Following a Plan

Agile Activities Triangle

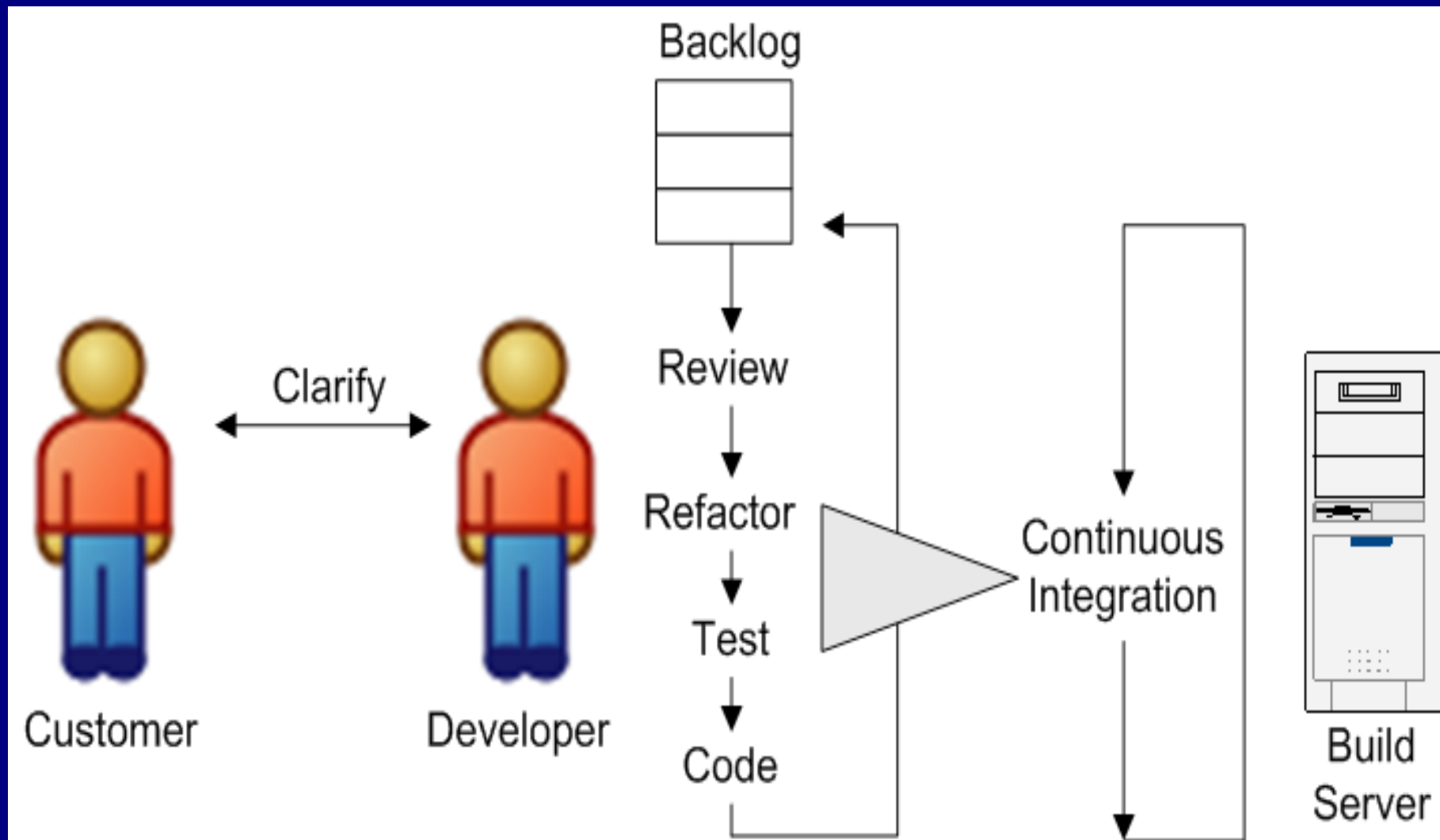


- Interdependent activities: All are needed to permit short adaptive iterations
- Removing one causes a collapse

Refactoring – Improve the existing code without changing its behavior

Continuous Integration – Frequent build and test of full system

Construction Iteration Activities

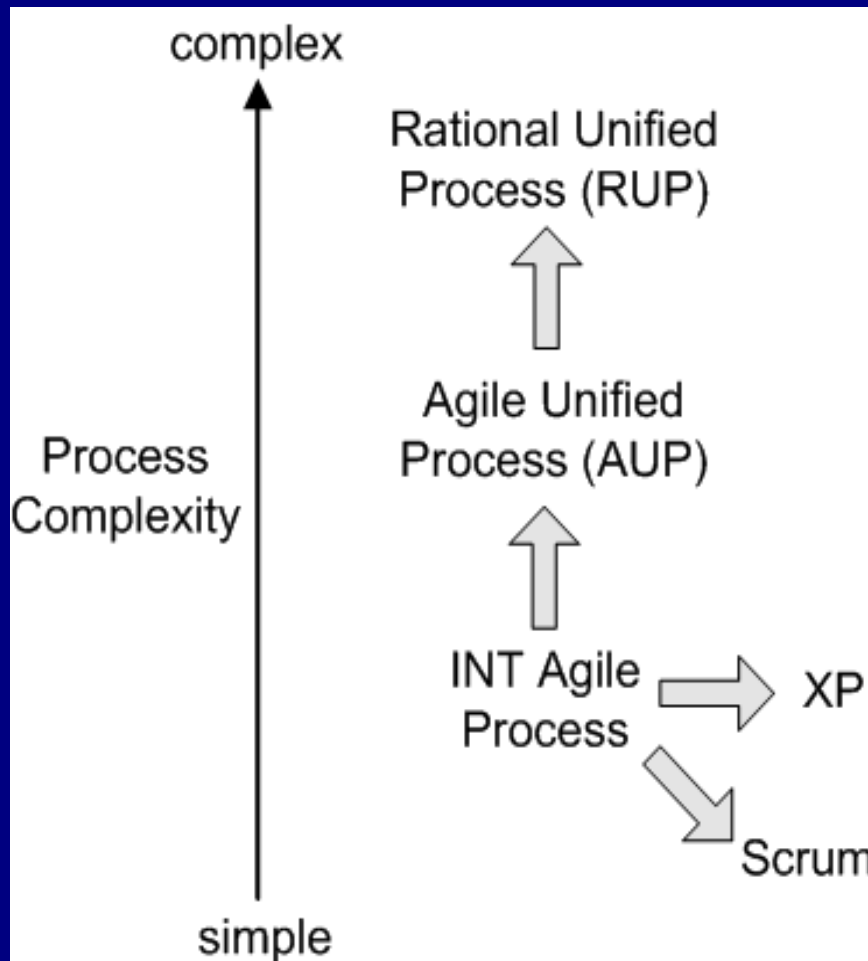


- The Backlog contains user stories
- Refactor any time the code smells and the tests pass

Use Spike Solutions

- Use a spike solution any time risk or uncertainty is high
- A spike solution is a very narrow but tall solution that explores some issue
 - *narrow*: only explores a particular issue
 - *tall*: implements all the software layers pertinent to the issue
- Spikes are only used when the issue is important to the customer

Relationship to Other Processes

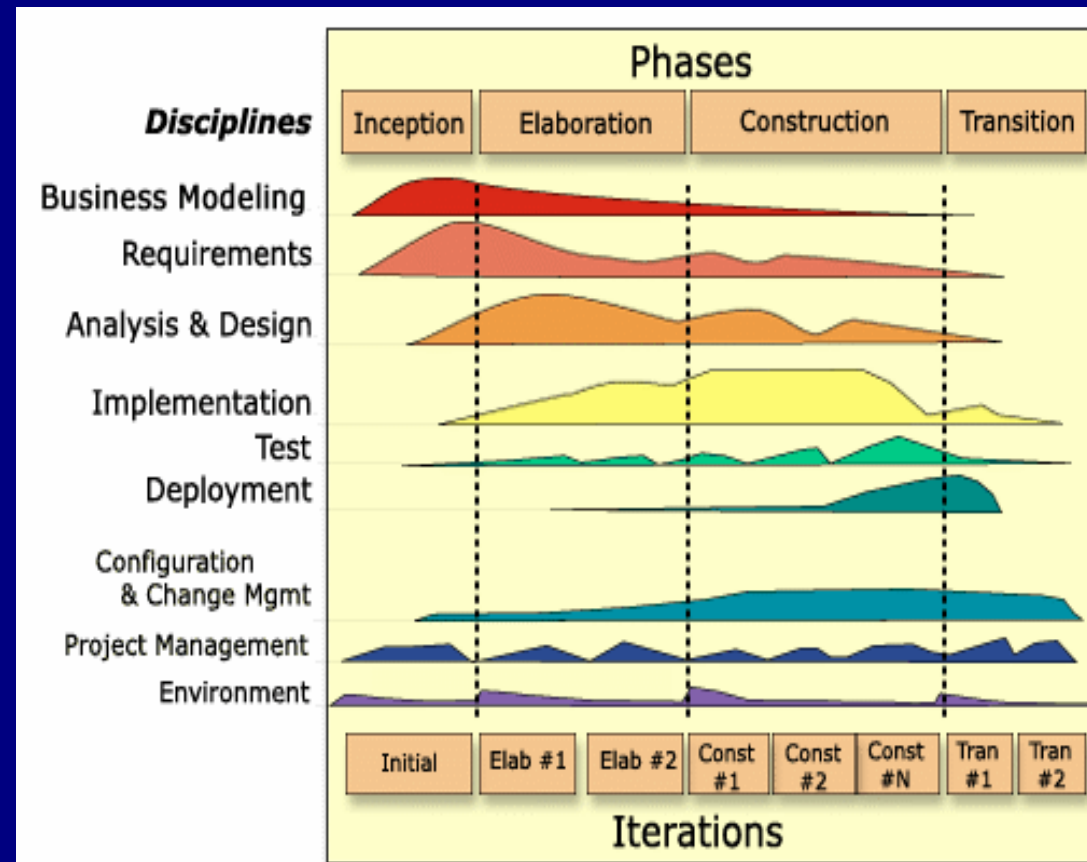


RUP, AUP and INT Agile Process have the same terminology for phases

- INT started with RUP, and is moving to more agile processes

Phases in the Process

- Inception – Establish a common vision (stories, business case & scope)
- Elaboration – Create a development plan, optionally build spikes
- 3. Construction – Build remainder of system
- 4. Transition – Deploy

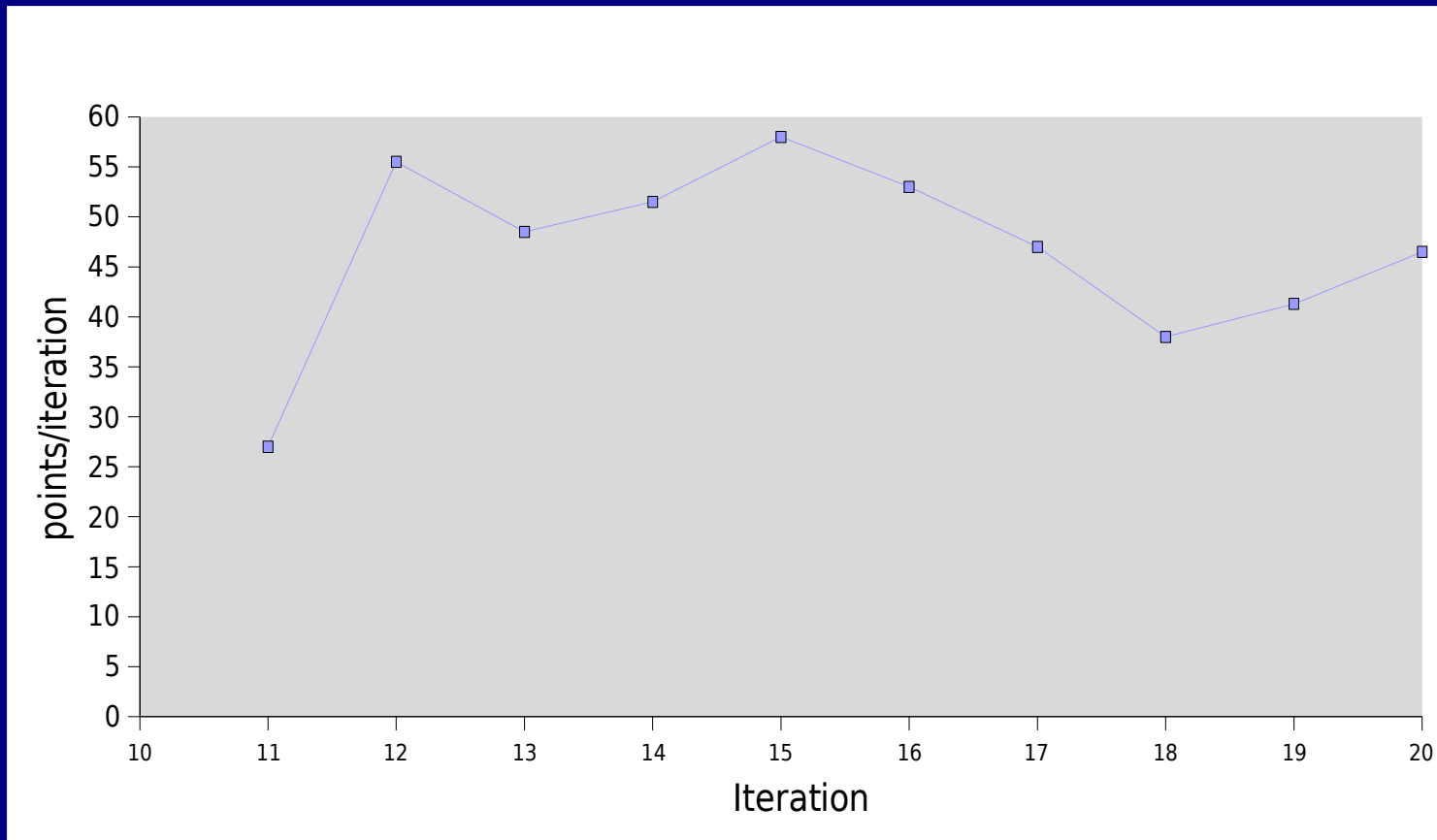


First Phases in INT Agile

- Inception
 - Customers meet with developers
 - Define project scope
 - Gather user stories
- Elaboration
 - Make spikes where needed
 - Estimate stories
 - Prioritize stories
 - Create a Development Plan

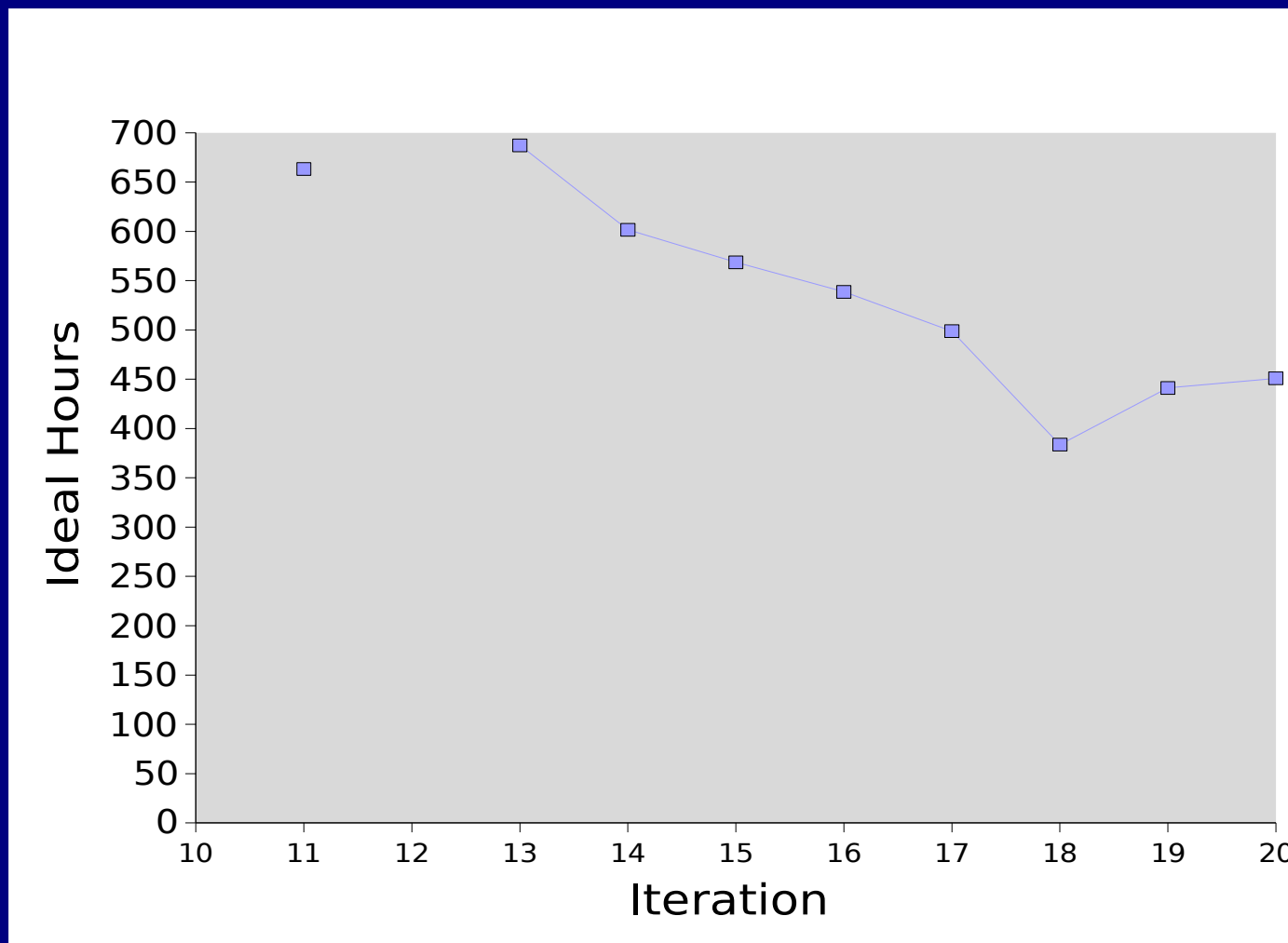
Project Management

- Stories estimates are relative to other stories in points
- Track the number of points completed in each iteration:
Velocity



Project Management

- Work not yet completed – Burndown Chart



Roles and Responsibilities

- Customer
 - Creates, modifies, and removes user stories
 - Assigns priorities to user stories
 - Works with Developers to create acceptance tests
 - Evaluates iteration releases and provides feedback

Roles and Responsibilities

- Developer
 - Estimates user stories
 - Implements user stories and unit tests
 - Creates and refactors system design
 - Writes documentation
 - Works with Customers to create acceptance tests
 - Plans iteration

Roles and Responsibilities

- Project Lead
 - Performs all developer duties
 - Gathers feedback from Customers and Developers
 - Facilitates Developer training
 - Monitors metrics (velocity, burndown chart, code coverage)
 - Reports to stakeholders