



Small is Beautiful

The Evolution of Agile to Complex, Multi-Project Environments



Sanjiv Augustine, LitheSpeed

Seattle, WA

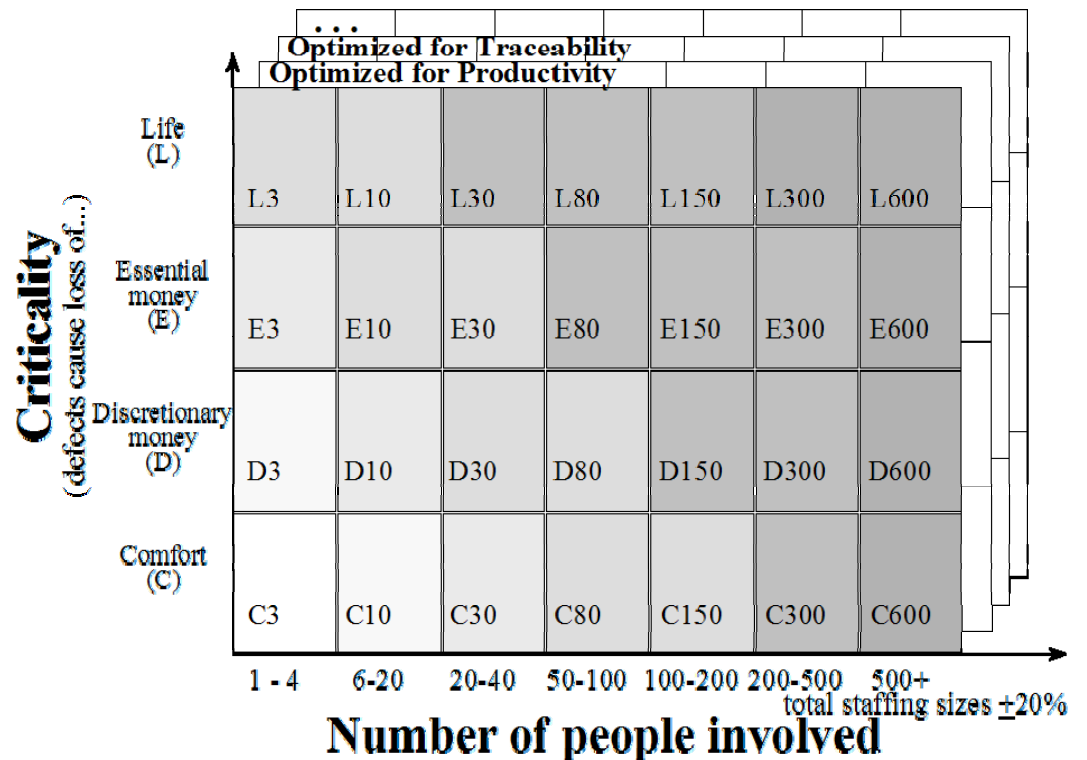
August 06, 2007

Aspect\Level	L1: Basic	L2: Evolved
<i>Small Teams</i>	Integrated Agile Team	Federation of Agile Teams
	Self-Organization	Self-Management
<i>Small Batches</i>	Individual Projects	Lean Project Portfolio
	Individual Products	Product Lines



Historical Sweet Spot:

- Small team (<10-15 people)
- Green field, custom development
- Collocated team members
- On-site customer
- Supporting technology/development environment
- Automated builds
- Automated unit testing
- Automated system testing
- Automated acceptance testing



Source: Alistair Cockburn

How can we evolve Agile to complex, multi-project environments?



Small is Beautiful Principle (1973)

- Developed by the German-born British economist and businessman ERNST SCHUMACHER (1911-1977), who advocated 'intermediate' and alternative technologies.
- Schumacher proposed the idea of "smallness within bigness"; in other words, a specific form of decentralization: ***for a large organization to work it must behave like a related group of small organizations.***
- Challenged the tradition of large organizations, which Schumacher claimed were inefficient and a danger to the environment. He proposed small working units, communal ownership, and the use of local labor and resources. He placed the emphasis on people rather than the product.

Sources: <http://www.economyprofessor.com/economictheories/small-is-beautiful-principle.php>;
http://en.wikipedia.org/wiki/Small_is_Beautiful

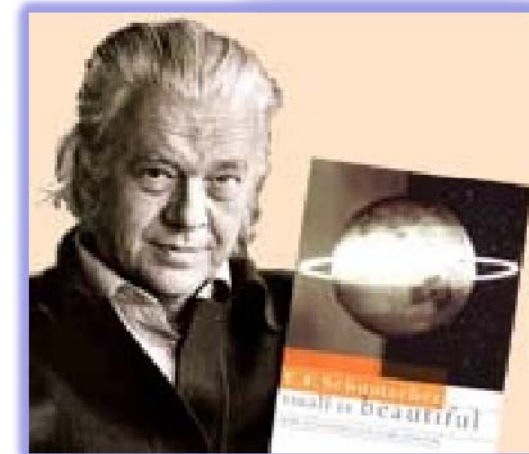


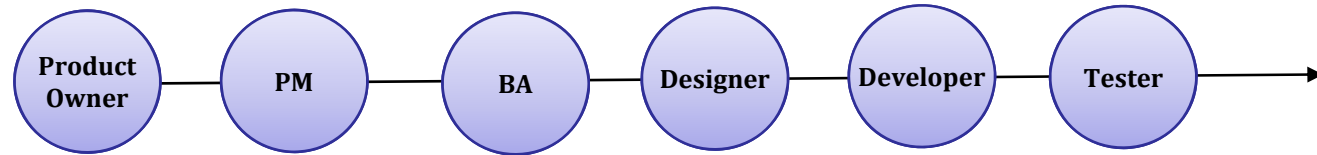
Image sources:
http://practicalactionconsulting.org/?id=is_small_beautiful; <http://www.iacknicholson.org/inphotos.html>



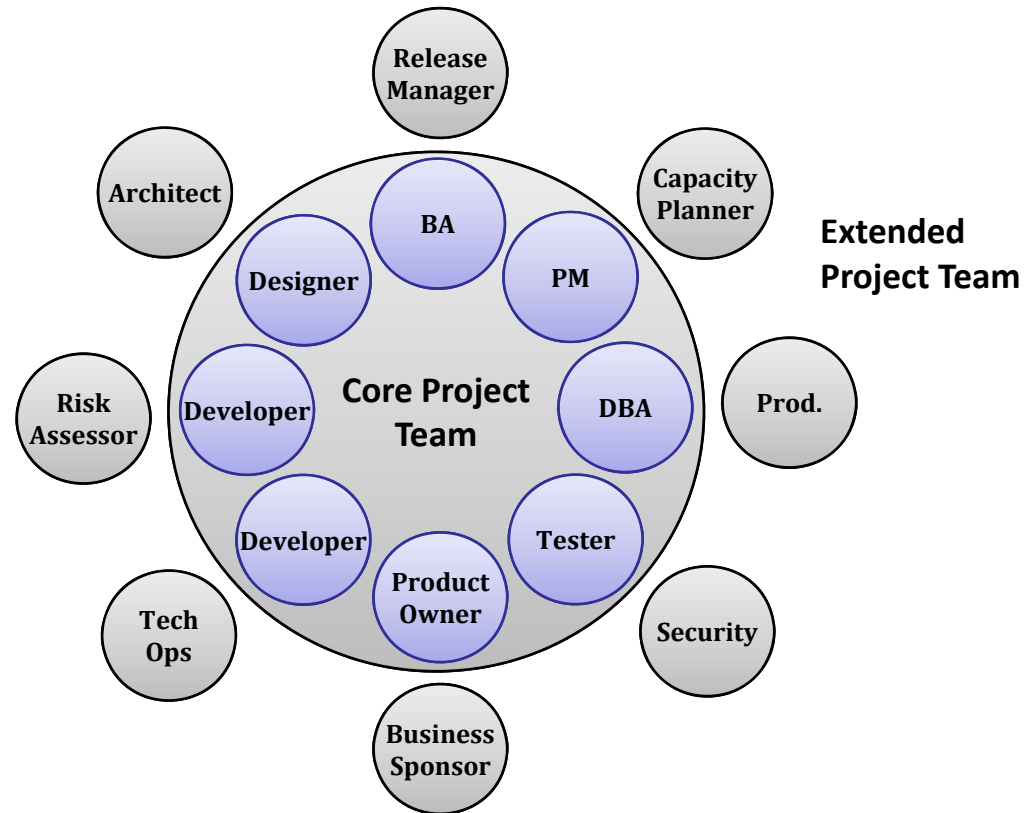


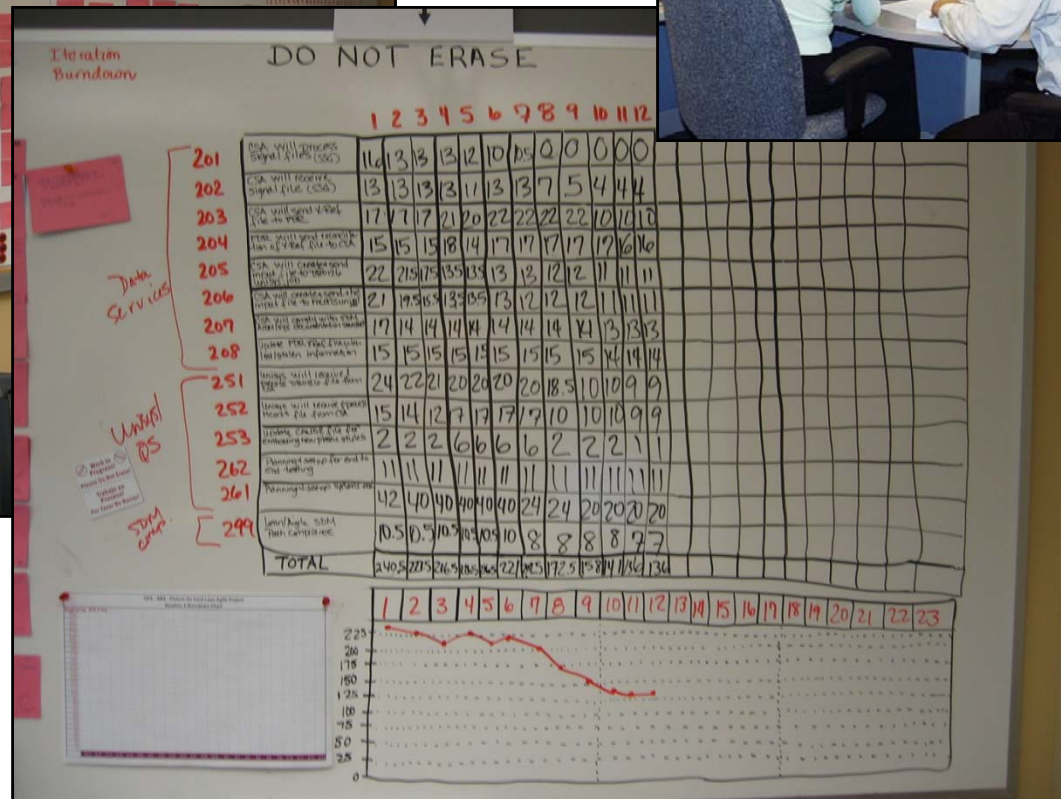
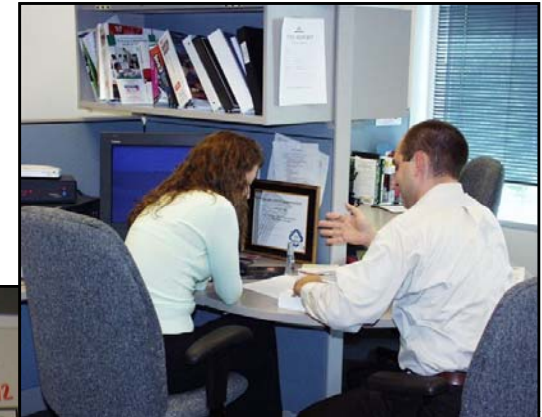
Small Teams

Traditional Silos:



Integrated Agile Team:

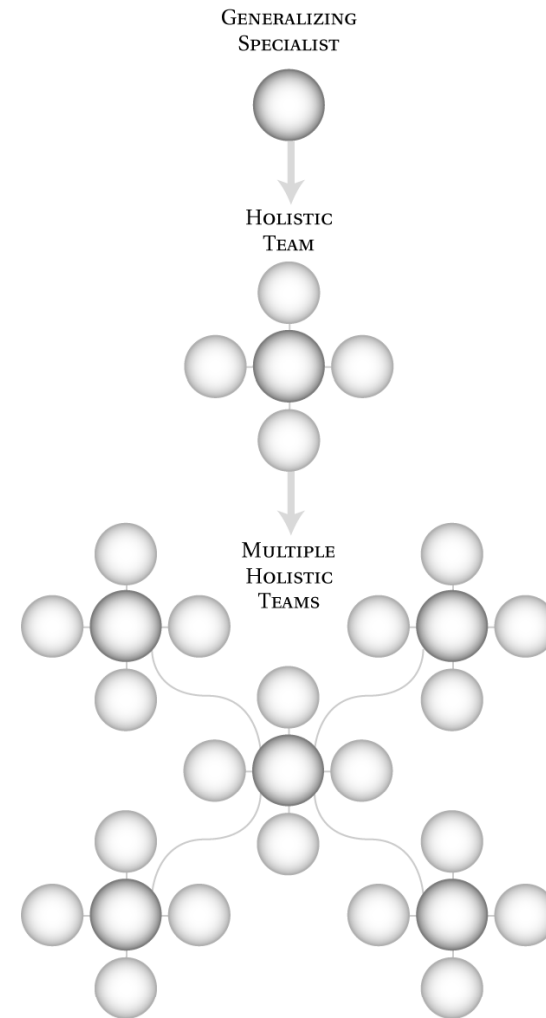


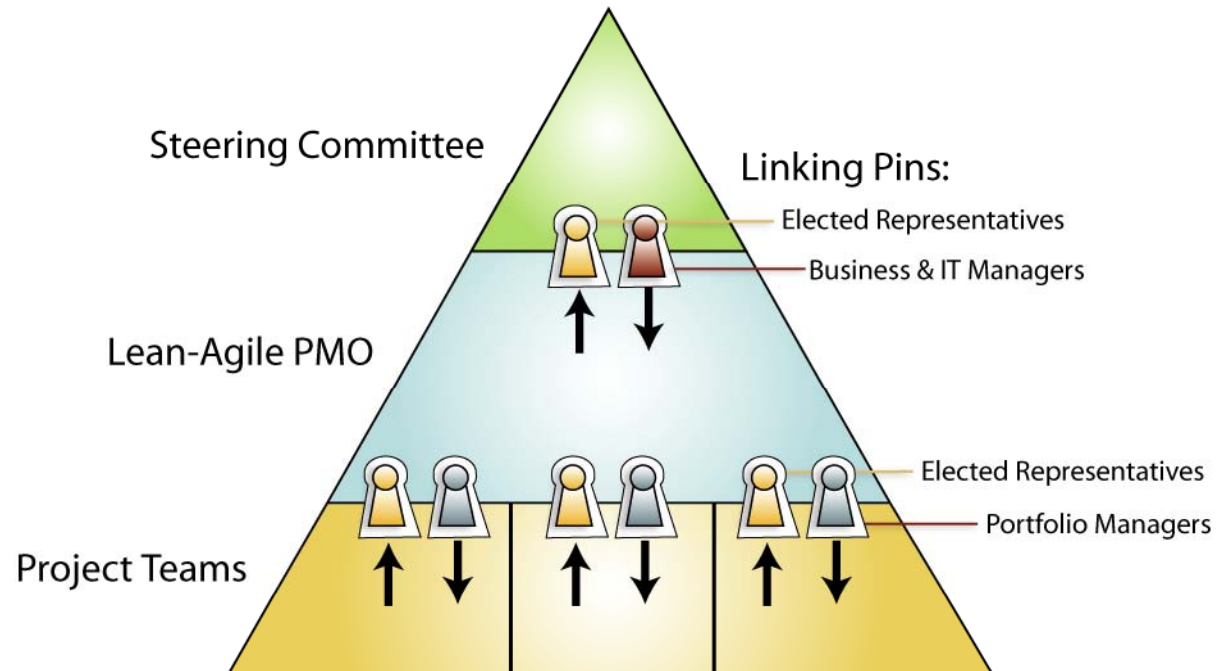


“...for a large organization to work it must behave like a related group of small organizations.”

Building a “federation of agile teams:”

- Build the “whole” into the “parts”
- Set a size limit (E.g. in Scrum 7+-2 people)
- To grow, break off new integrated Agile teams when team size limit is reached
- Coordinate at higher level via Lean-Agile PMO





- Encourage face-to-face dialogue across levels
- Overlapping management with “linking pins”
- Lean-Agile PMO run as an Agile project team

Source: *The Lean-Agile PMO*, Sanjiv Augustine and Roland Cuellar (Cutter Consortium 2006)



Self-organization refers to a process in which the internal organization of a system, normally an open system, increases automatically without being guided or managed by an outside source.

Wikipedia

Self-organizing teams:

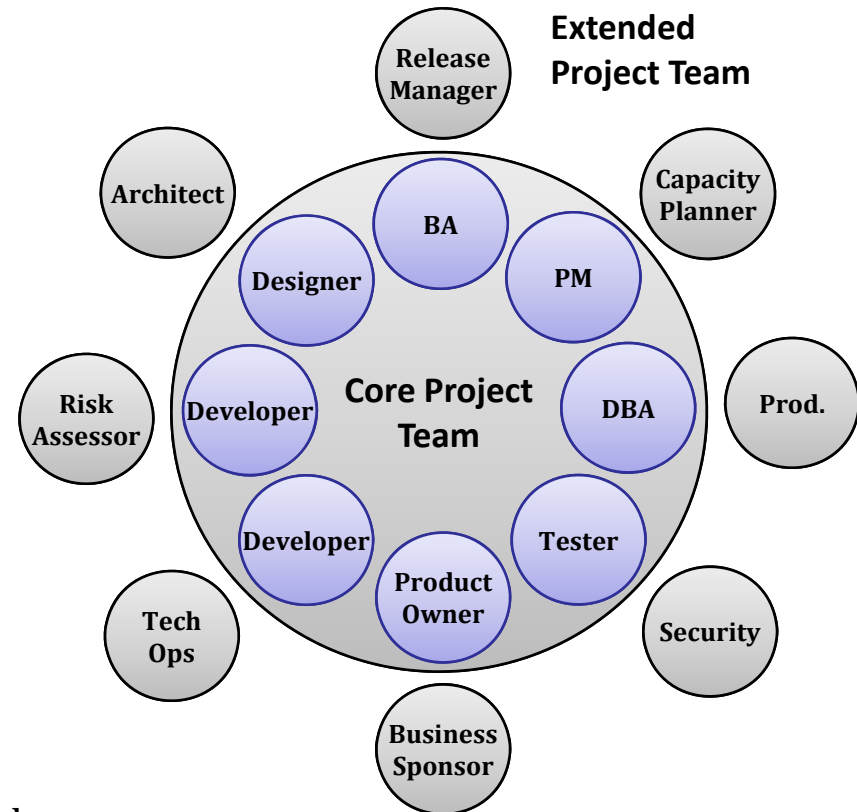
- Exhibit a high degree of collaboration
- Operate with a high degree of trust and autonomy
- Work towards very high performance
- Produce measurably great results
- Are very fulfilling to work on

Characteristics of Agile Self-Organizing Teams

- Small team size
- Customer value orientation
- Individual competence
- Sustainable self-discipline
- Intense collaboration
- Easy information transfer
- Low decision feedback time
- Constant learning and interaction



- Small integrated team
- Clear mission
- Short timeframe
- Proper leadership
- All necessary roles:
 - ScrumMaster/PM
 - Business Analyst
 - Tester
 - Developer
 - Data Designer/DBA
- Good understanding of customer needs
- Assured of getting needed resources
- Empowered to take action
- Basic engineering and other disciplines in place



Source: Mary and Tom Poppendieck



PETER DRUCKER



- Concentrate on your strengths
- Improve your strengths
- Overcome intellectual arrogance
- Avoid areas of weakness



Build on Personal Strengths.

- **Applying it to Others:**

- Each person is unique and has unique strengths and weaknesses
- Great managers recognize that trying to standardize human behavior is futile, and don't waste their time trying to change people dramatically
- Rather than focus on weaknesses, they build on the personal strengths of their team members and help them become more of "who they already are"

- **Applying it to Yourself:**

- Find out what you don't like doing and stop doing it
- "The point is to feel authentic, self-assured or creative" <http://www.marcusbuckingham.com>
- Conduct *feedback analysis*



“Productivity” is *not* a mechanical measure of speed.

- Knowledge drives productivity
- Knowledge workers need to own the responsibility for their own productivity
- Continuous innovation, learning and teaching need to be part of the job
- Knowledge worker productivity is dependent on *quality* at least as much as *quantity*
 - Optimal quality is the path to high productivity



- 6x Honda's **President Award** winner
 - “Most prestigious and coveted honor that Honda can grant to its dealerships. The award recognizes those dealership teams that demonstrate superior achievement in customer satisfaction, new-car-unit sales volume and business management. Approximately 15% of Honda dealerships achieve the President’s Award status each year. “
 - “President's Award recipients realize what this honor demands - excellence, which can only be achieved by working as a team. “
 - “The President's Award reflects a total commitment to customers; it is the highest award that a Honda dealership can receive because *it demands the highest personal investment and stimulates the greatest results - satisfied, loyal customers.*”

- “Although some of them seem to rate our entire operation, **in fact they all rate my personal performance and the scores become part of my permanent record.**”

Landmark Honda



I know you had a choice of where to service your vehicle and I would like to thank you for the opportunity to earn your business. I also know that if I expect to build a strong customer base, I need to make sure that you are completely satisfied with your overall service experience here today.

Listed below are some questions that may appear on a questionnaire from my dealership or possibly even a survey from Honda. Although some of them seem to rate our entire operation, in fact they all rate my personal performance and the scores become part of my permanent record. If you feel that I have not earned all “Excellent”, or “Yes” ratings, please call and allow me the opportunity to correct any of my failures or shortcomings. While my score is very important, my desire to earn and keep your business is far more important. If there is any situation where I may be of assistance now or in the future, please let me know.

	Pass	Fail	→
	Excellent	Good	Poor
Promptness and courtesy in greeting you and writing your order	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Service advisor's knowledge and expertise	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Survey



- **How do we define quality?**
 - Largely a matter of defining the work at hand
- **Knowledge workers need to first define:**
 - What is our business?
 - Who is our customer?
 - What does our customer consider valuable?

How to Evolve Productivity Metrics?

- Bugs reported in production
- Time to repair production bugs

Measures of quality must flow from the above.



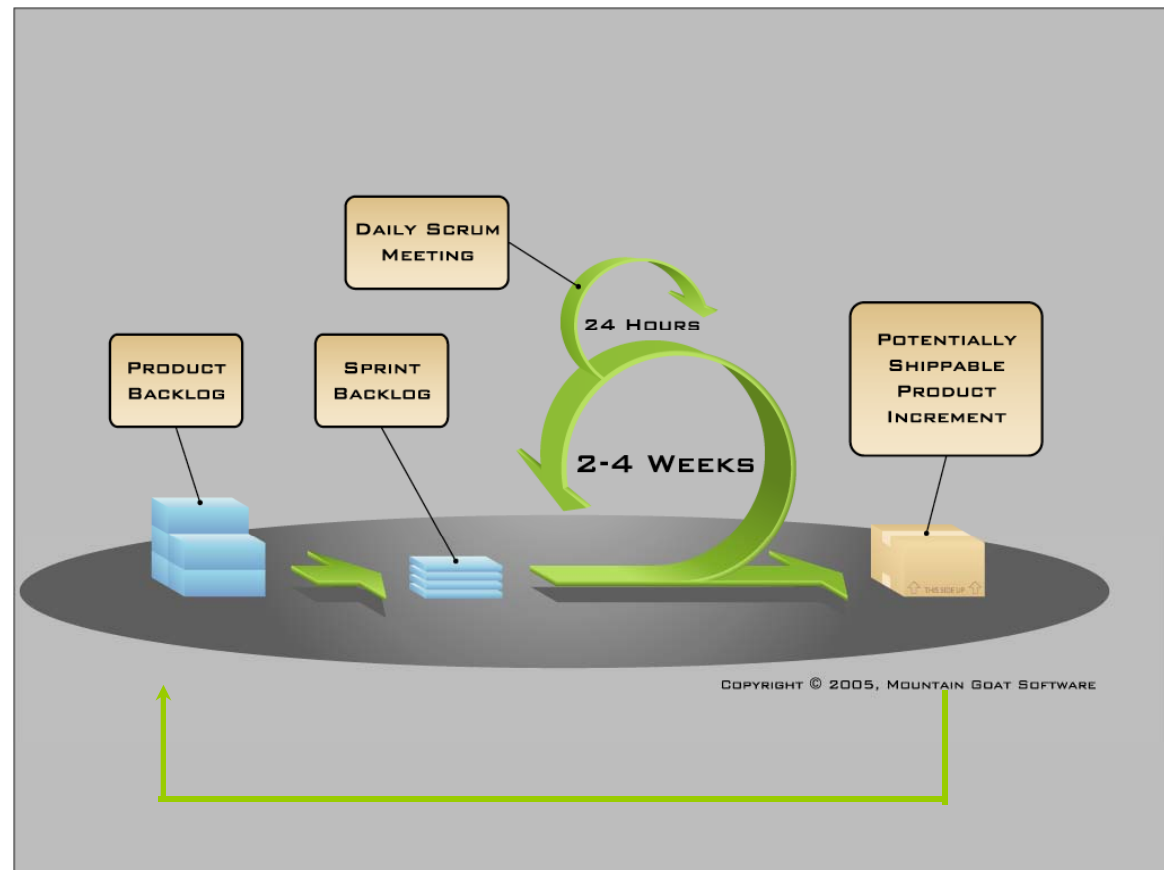


Small Batches

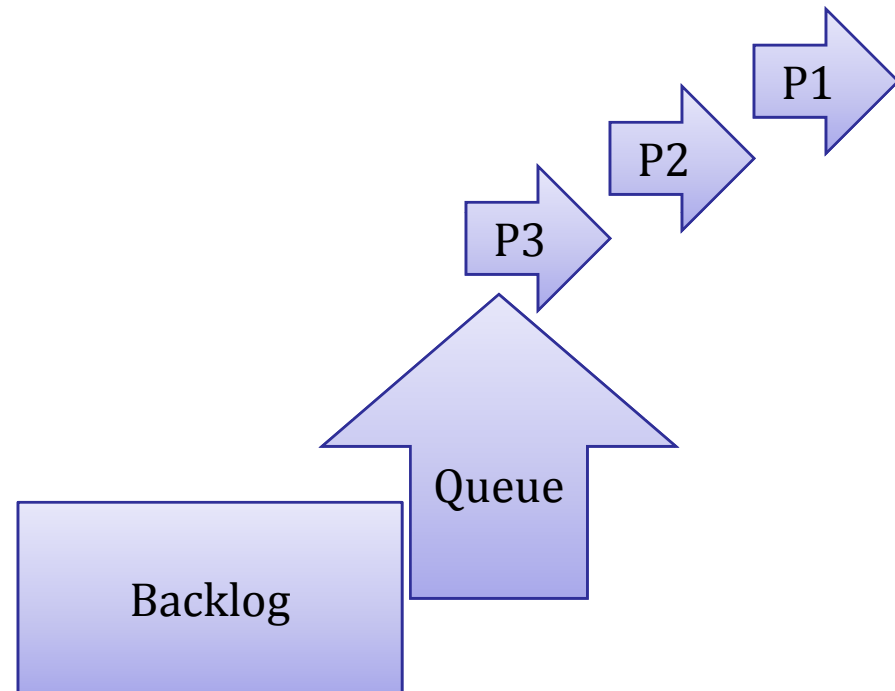
Key Agile Practices:

- Release Planning
- Sprint Planning
- Daily Scrum/Standup
- Fixed-length sprints
- Sprint Review
- Sprint Retrospective

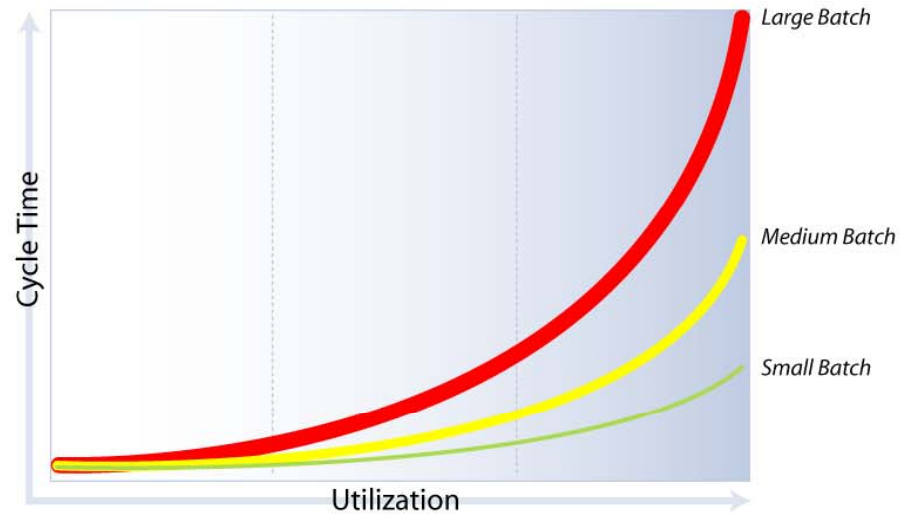
Identify top-priority items and deliver them rapidly in small batches.



- **Some principles to manage small “project batches”**
 - Optimize for Throughput
 - Reduce Project Inventory/WIP
- **Applying the Principles**
 - Portfolio Re-Alignment
 - Lean Resource Management
 - Platform Based Teams



- Optimal utilization yields sub-optimal throughput
- “There is no free lunch”
- Delay has a cost



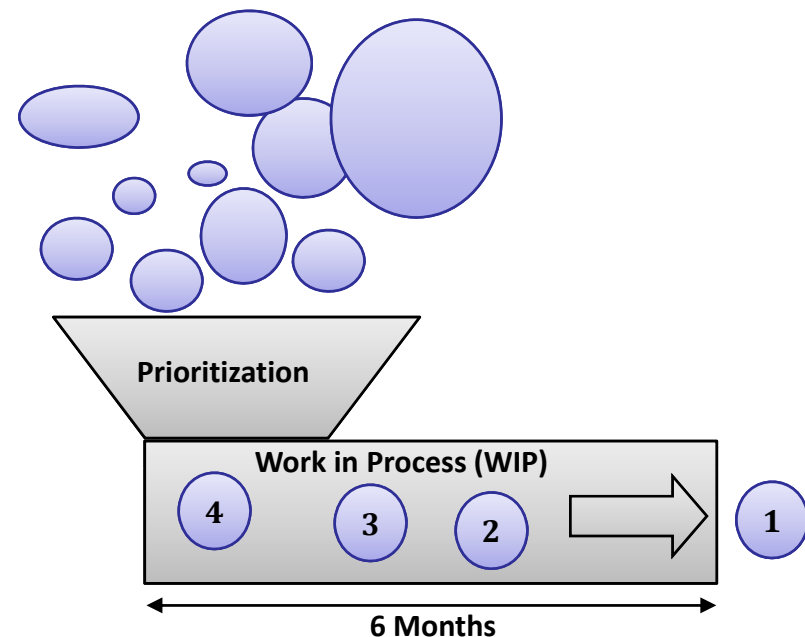
Source: *The Lean-Agile PMO*, Sanjiv Augustine and Roland Cuellar (Cutter Consortium 2006)



- Projects are the “inventory” in the portfolio
- **Big inventories are bad:**
 - They cost a lot
 - They hide a lot of problems
 - They are slow



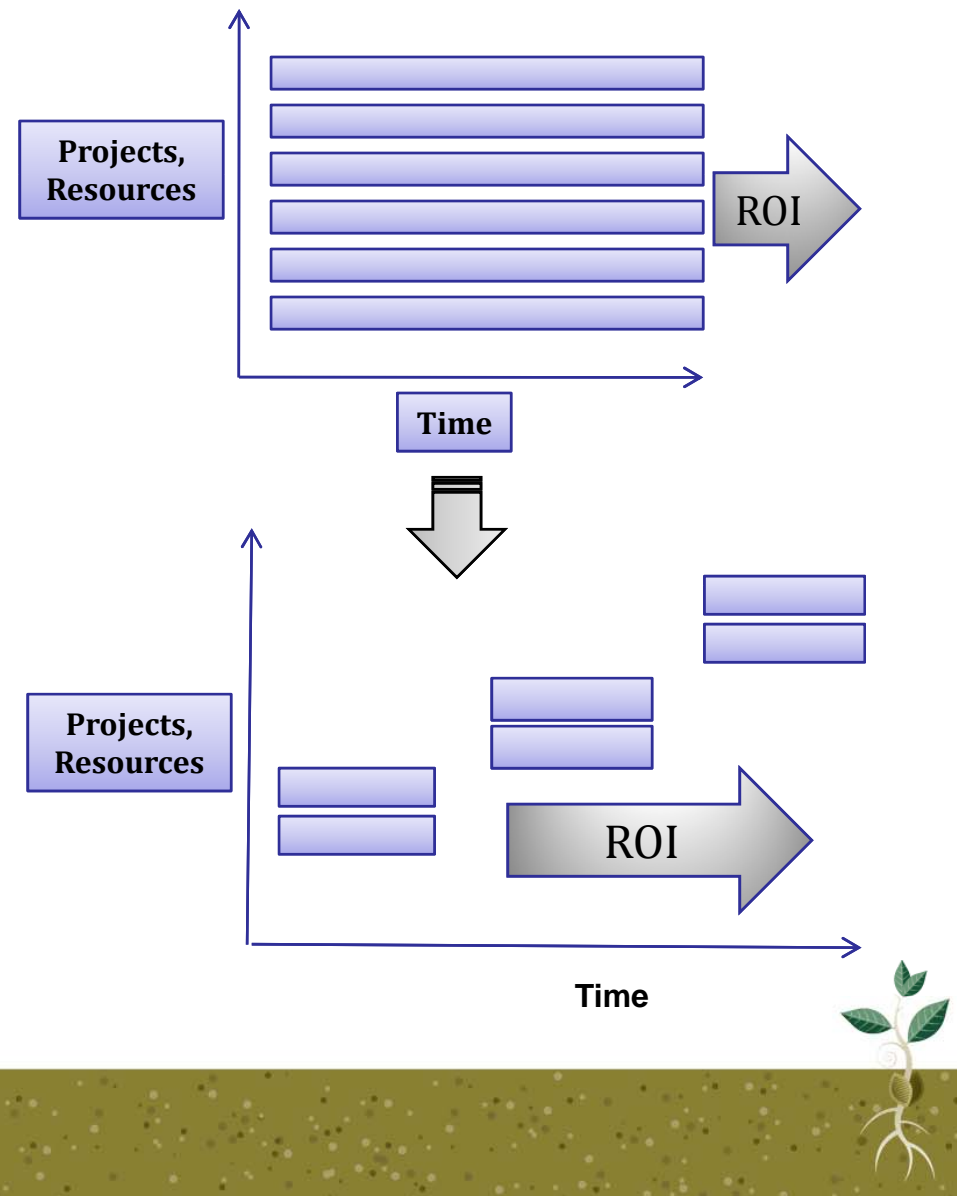
- Terminate sick projects
- Split large projects in smaller ones
- Prioritize projects by business value, at least within business unit
- Limit delivery timeframe to months
- Re-prioritize projects regularly



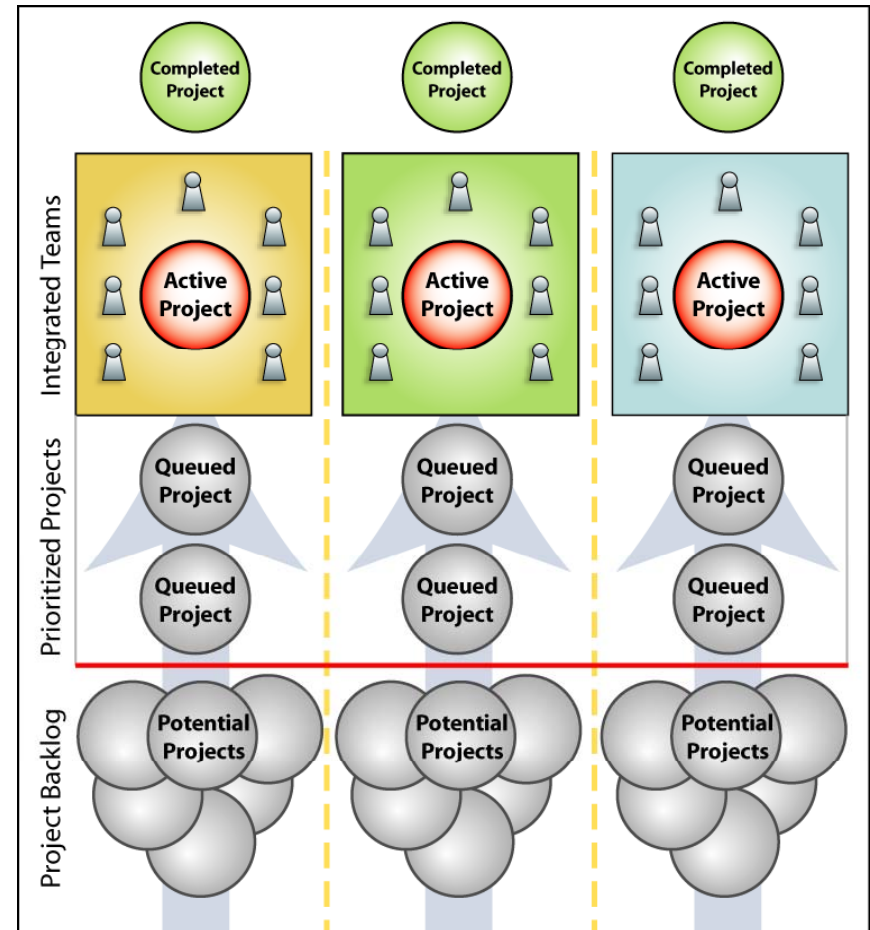
$$\text{Cycle Time} = \frac{\text{WIP}}{\text{Completion Rate}}$$



- Dedicate core resources to each project team
- Ensure that each team has all resources needed to complete the project
- Stress maximum project throughput
- ROI delivered incrementally with each project release



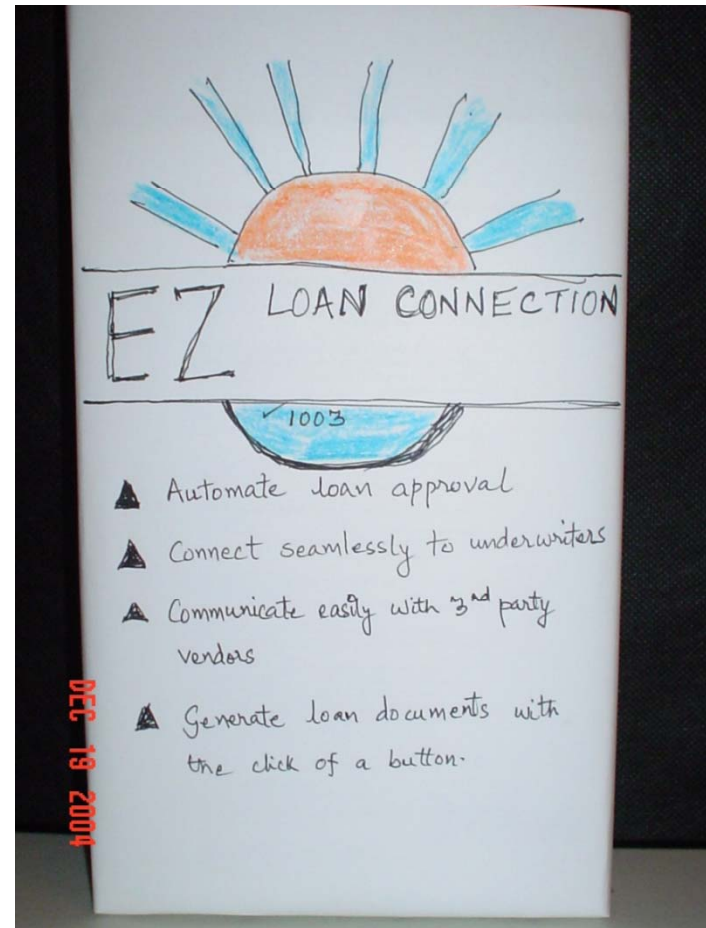
- Multiple teams each focused on a single project
- Dedicated to platforms or lines of business
- Platform owner prioritizes next project
- Result:
 - Support multiple lines of business simultaneously
 - Focused effort results in quick delivery for individual projects
 - Clear accountability



Source: *The Lean-Agile PMO*, Sanjiv Augustine and Roland Cuellar (Cutter Consortium 2006)



- Shared project and product vision, with standalone products
- Product Owner as creator/arbiter of product vision
- Incremental product development and rollout



Source: *Managing Agile Projects*, Sanjiv Augustine



- Engineering techniques for creating a portfolio of similar software systems from a shared set of software assets using a common means of production
- Strategic software reuse – predictive versus opportunistic software reuse.
- Software artifacts are created when reuse is predicted in one or more products in a well defined product line
- Common core components are developed and shared quickly across multiple projects.
- Enables *mass customization* - creating many variations of a product by capitalizing on their commonality

Source:<http://www.softwareproductlines.com/introduction/introduction.html>

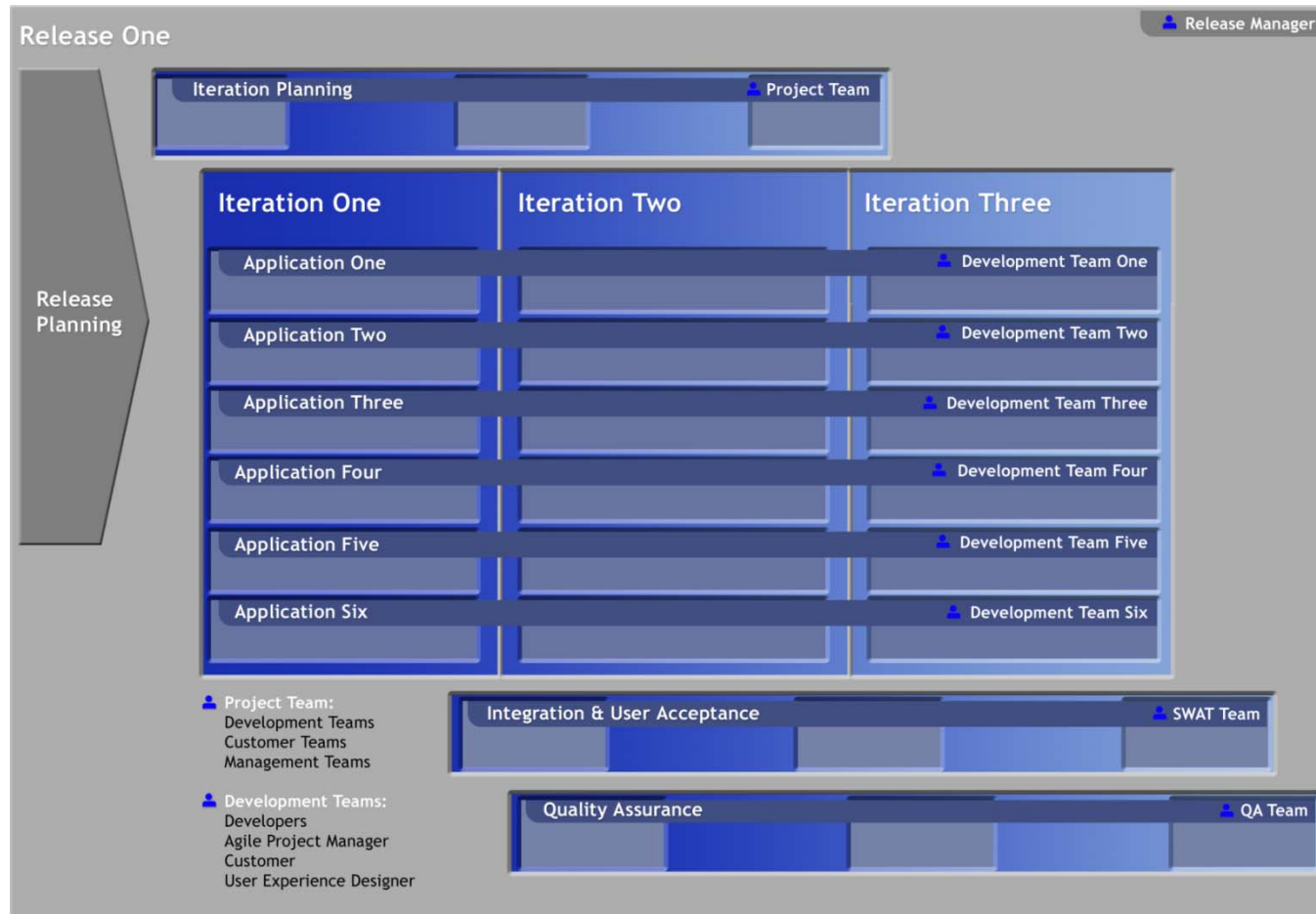


Diesel Engine Software Product Line, Cummins, Inc.

- Modern engines can contain over 100KSLOC of software to micro-control ignition to produce an optimum mix of power, economy, and emissions.
- In 1993, faced with the need to produce almost 20 new systems but with staff and resources available only for six, Cummins embraced the product line approach.
- Today, the Cummins software product line covers 9 basic engine types ranging over 4-18 cylinders and 4-164 liters of displacement, with 12 kinds of electronic control modules, 5 kinds of processors, and 10 kinds of fuel systems.
- Cycle time has been reduced from around 250 person months to a few person months. Quality and customer satisfaction are both up, and 15 of 15 projects are on track.
- To date, 20 basic software builds have been parlayed into well over 1000 separate products.

Source: http://www.sei.cmu.edu/productlines/plp_hof.html





Source: *Steering Projects with Agile Project Management*, Sanjiv Augustine et al





Contact Us for Further Information

Sanjiv Augustine

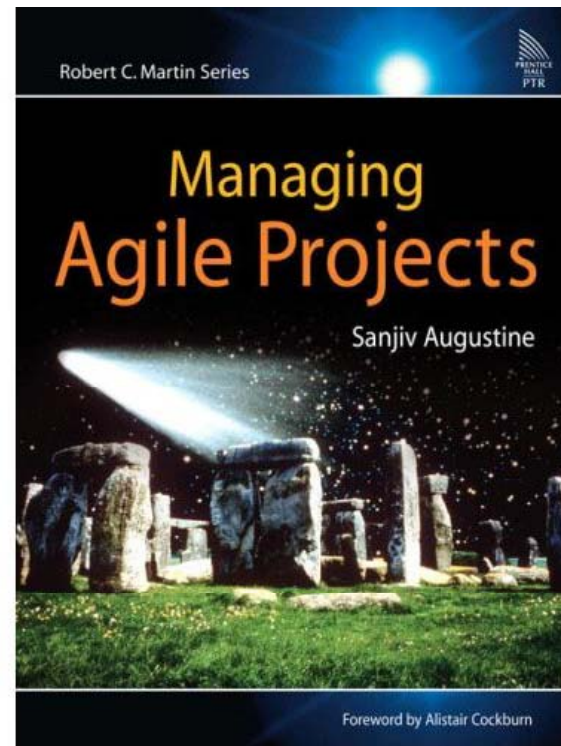
President

Sanjiv.Augustine@lithespeed.com

Sanjiv@sanjivaugustine.com

<http://www.lithespeed.com>

<http://www.sanjivaugustine.com>



"I only wish I had read this book when I started my career in software product management, or even better yet, when I was given my first project to manage. In addition to providing an excellent handbook for managing with agile software development methodologies, *Managing Agile Projects* offers a guide to more effective project management in many business settings."
John P. Barnes, former Vice President of Product Management at Emergis, Inc.



- Establish synchronized drop points
- Manage dependencies closely – monitor queues/backlogs
- Integrate liaison(s) into Agile teams and/or Lean-Agile PMO

