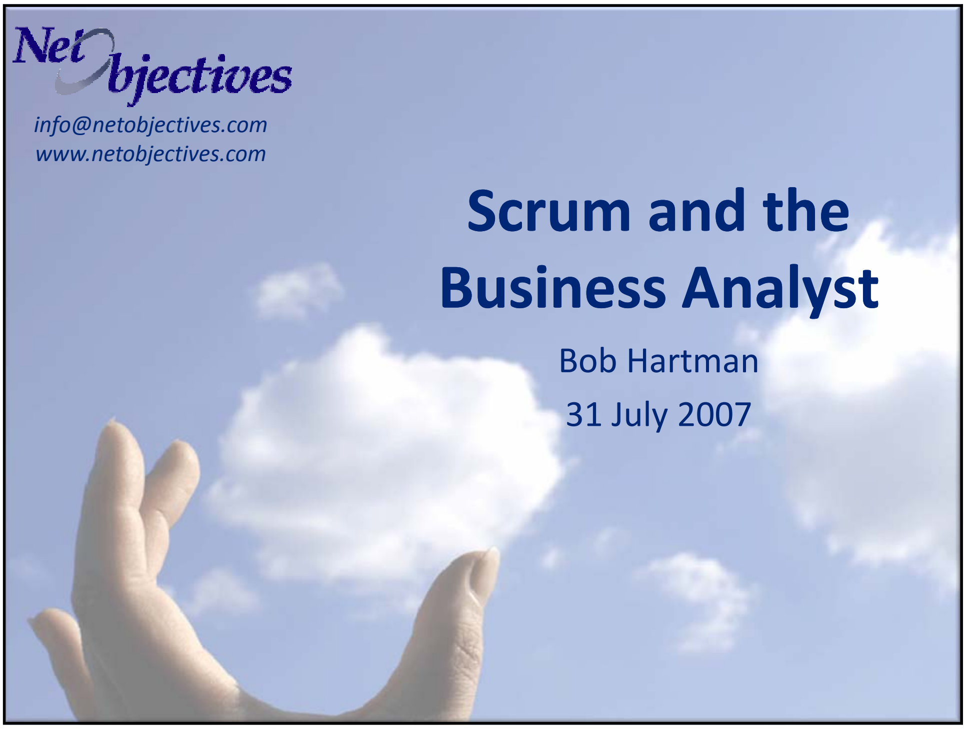
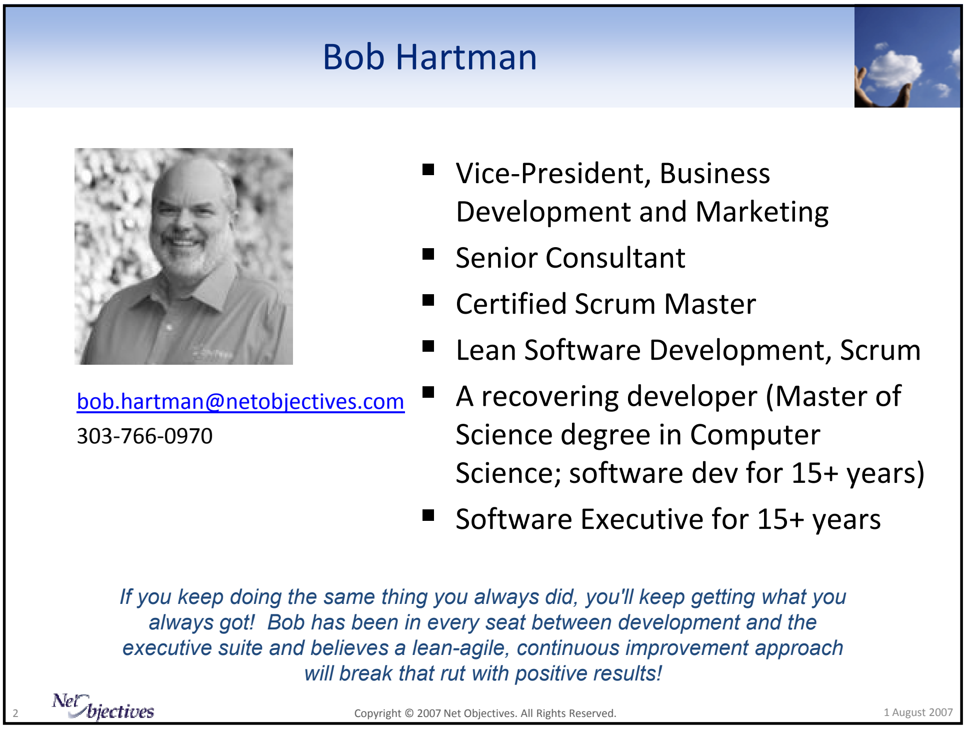


info@netobjectives.com
www.netobjectives.com

Scrum and the Business Analyst

Bob Hartman

31 July 2007

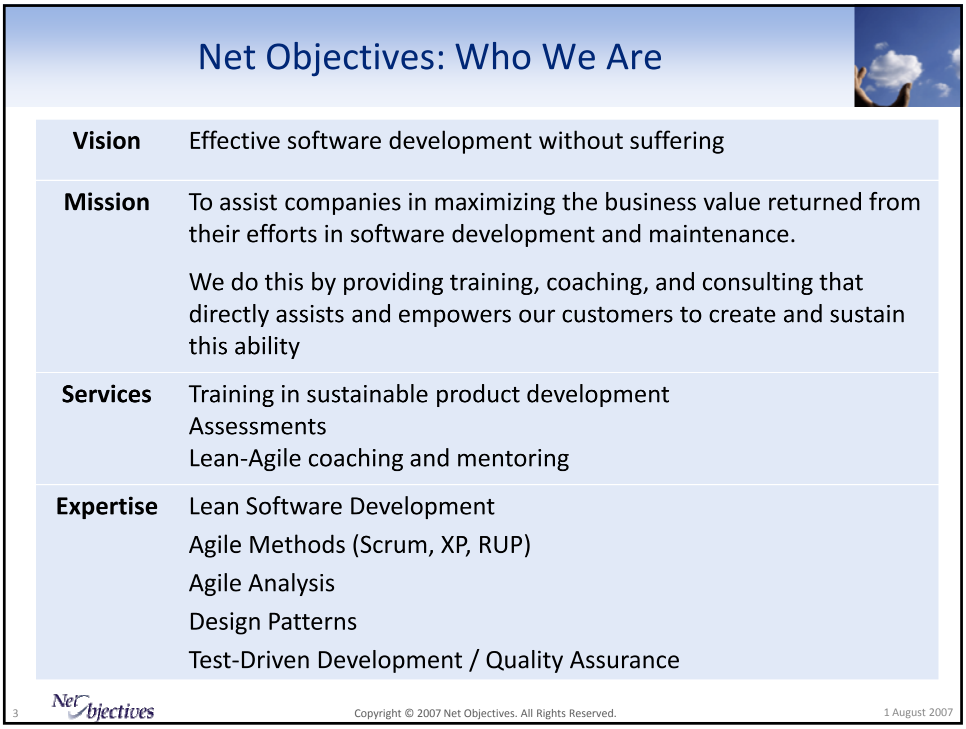


Bob Hartman

bob.hartman@netobjectives.com
303-766-0970

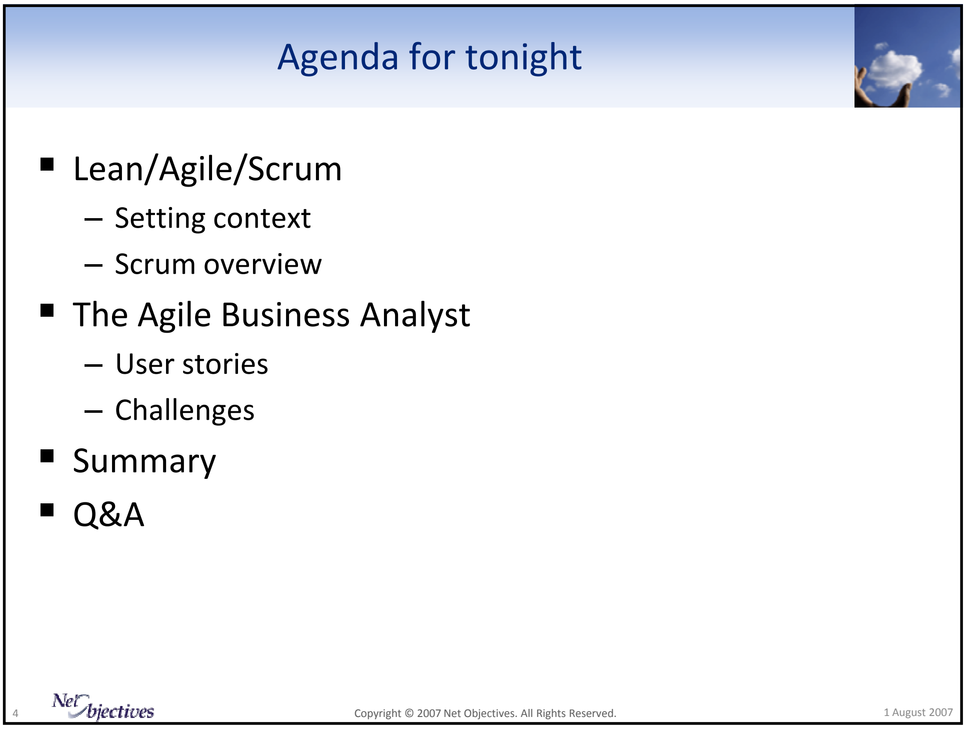
- Vice-President, Business Development and Marketing
- Senior Consultant
- Certified Scrum Master
- Lean Software Development, Scrum
- A recovering developer (Master of Science degree in Computer Science; software dev for 15+ years)
- Software Executive for 15+ years

If you keep doing the same thing you always did, you'll keep getting what you always got! Bob has been in every seat between development and the executive suite and believes a lean-agile, continuous improvement approach will break that rut with positive results!



Net Objectives: Who We Are

Vision	Effective software development without suffering
Mission	To assist companies in maximizing the business value returned from their efforts in software development and maintenance. We do this by providing training, coaching, and consulting that directly assists and empowers our customers to create and sustain this ability
Services	Training in sustainable product development Assessments Lean-Agile coaching and mentoring
Expertise	Lean Software Development Agile Methods (Scrum, XP, RUP) Agile Analysis Design Patterns Test-Driven Development / Quality Assurance



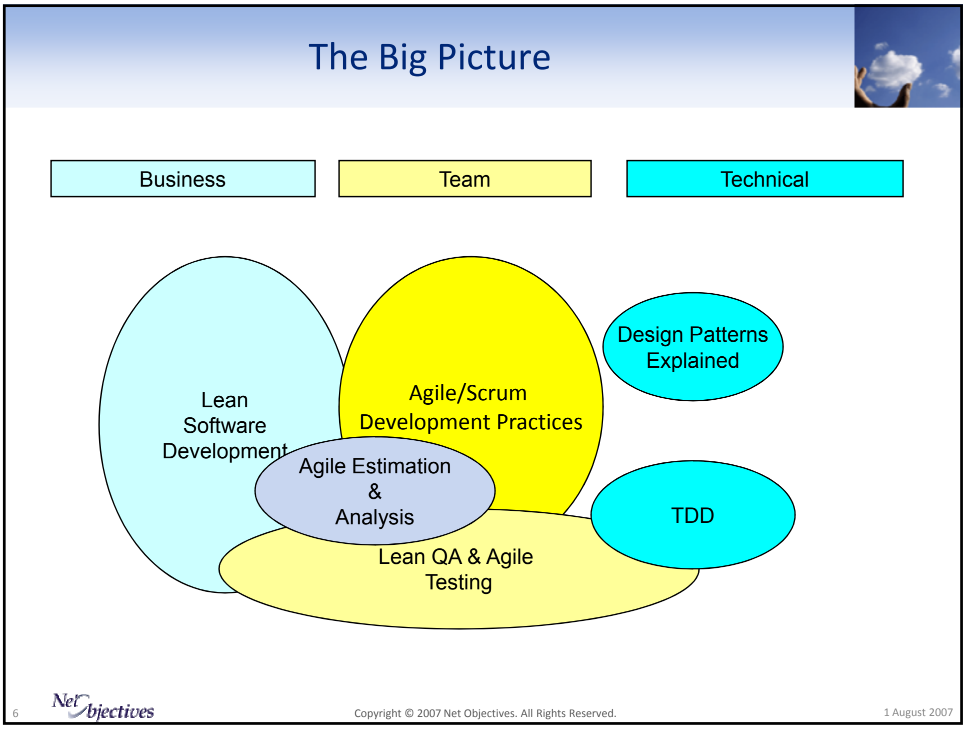
Agenda for tonight

- Lean/Agile/Scrum
  - Setting context
  - Scrum overview
- The Agile Business Analyst
  - User stories
  - Challenges
- Summary
- Q&A

Net Objectives

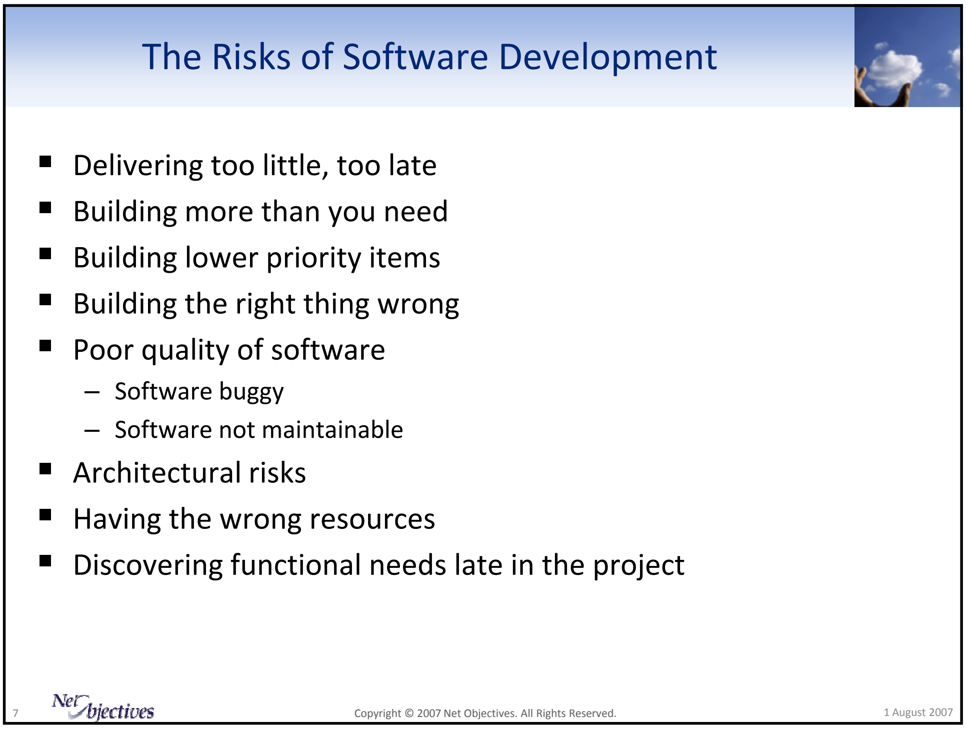
info@netobjectives.com
www.netobjectives.com

Setting Context



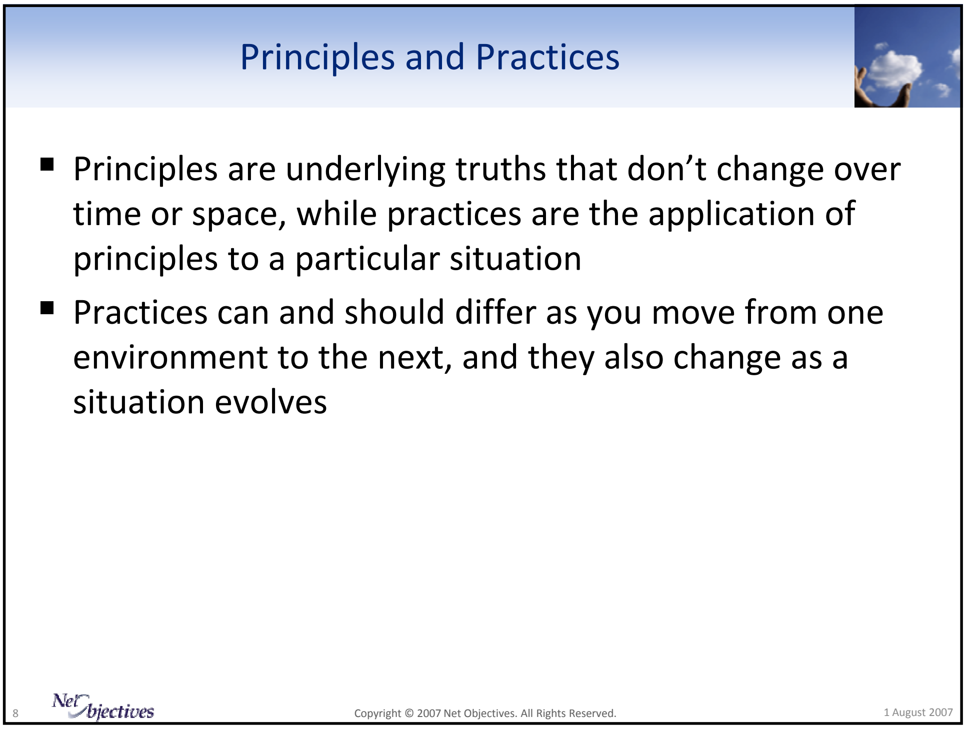
The Big Picture

Business
Team
Technical
Lean Software Development
Agile/Scrum Development Practices
Design Patterns Explained
Agile Estimation & Analysis
TDD
Lean QA & Agile Testing



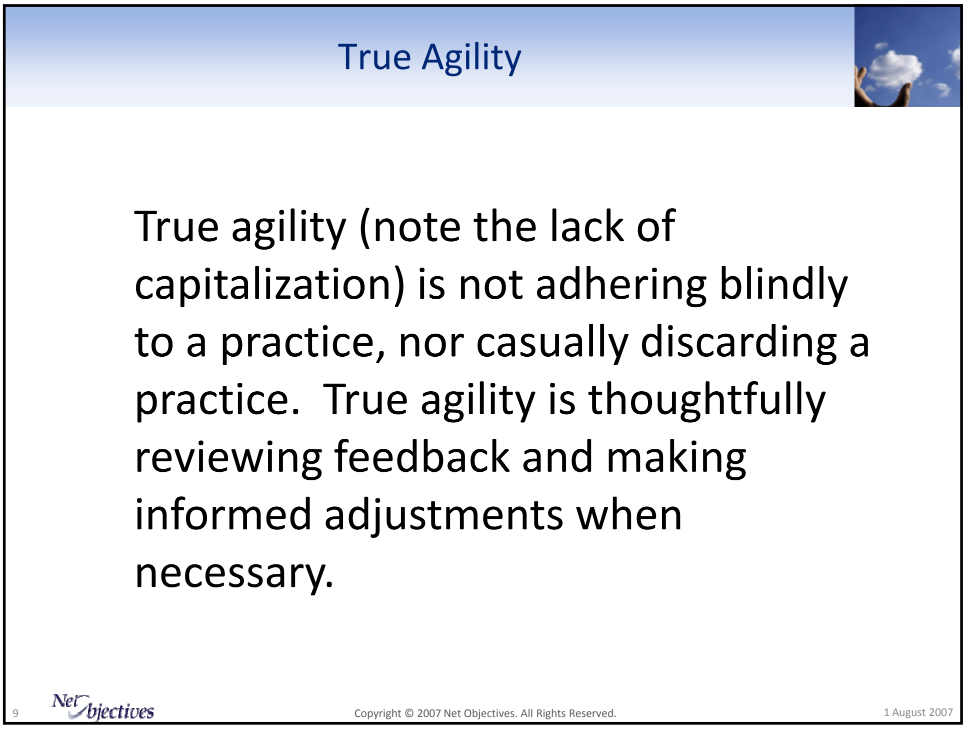
The Risks of Software Development

- Delivering too little, too late
- Building more than you need
- Building lower priority items
- Building the right thing wrong
- Poor quality of software
  - Software buggy
  - Software not maintainable
- Architectural risks
- Having the wrong resources
- Discovering functional needs late in the project



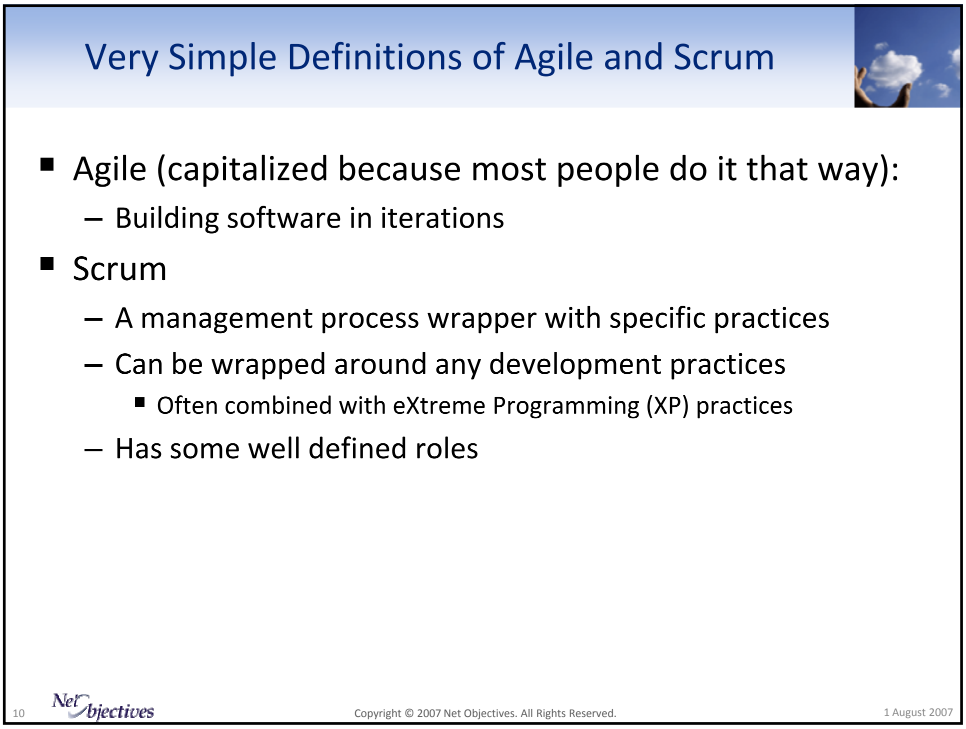
Principles and Practices

- Principles are underlying truths that don't change over time or space, while practices are the application of principles to a particular situation
- Practices can and should differ as you move from one environment to the next, and they also change as a situation evolves



True Agility

True agility (note the lack of capitalization) is not adhering blindly to a practice, nor casually discarding a practice. True agility is thoughtfully reviewing feedback and making informed adjustments when necessary.

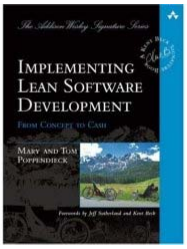


Very Simple Definitions of Agile and Scrum

- Agile (capitalized because most people do it that way):
  - Building software in iterations
- Scrum
  - A management process wrapper with specific practices
  - Can be wrapped around any development practices
    - Often combined with eXtreme Programming (XP) practices
  - Has some well defined roles

Principles of Lean Software Development

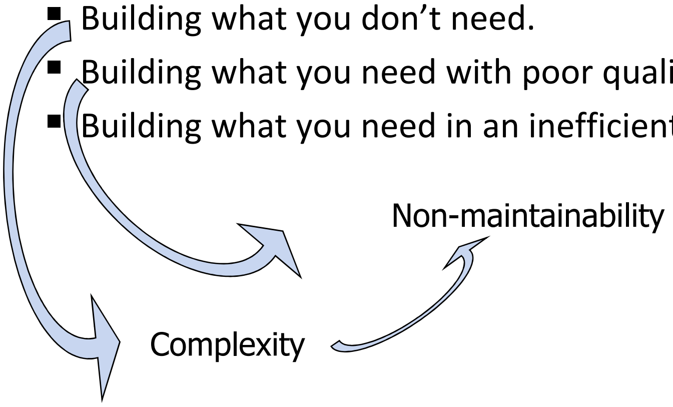
1. Eliminate Waste
2. Build Quality In
3. Create Knowledge
4. Defer Commitment
5. Deliver Fast
6. Respect People
7. Optimize the Whole

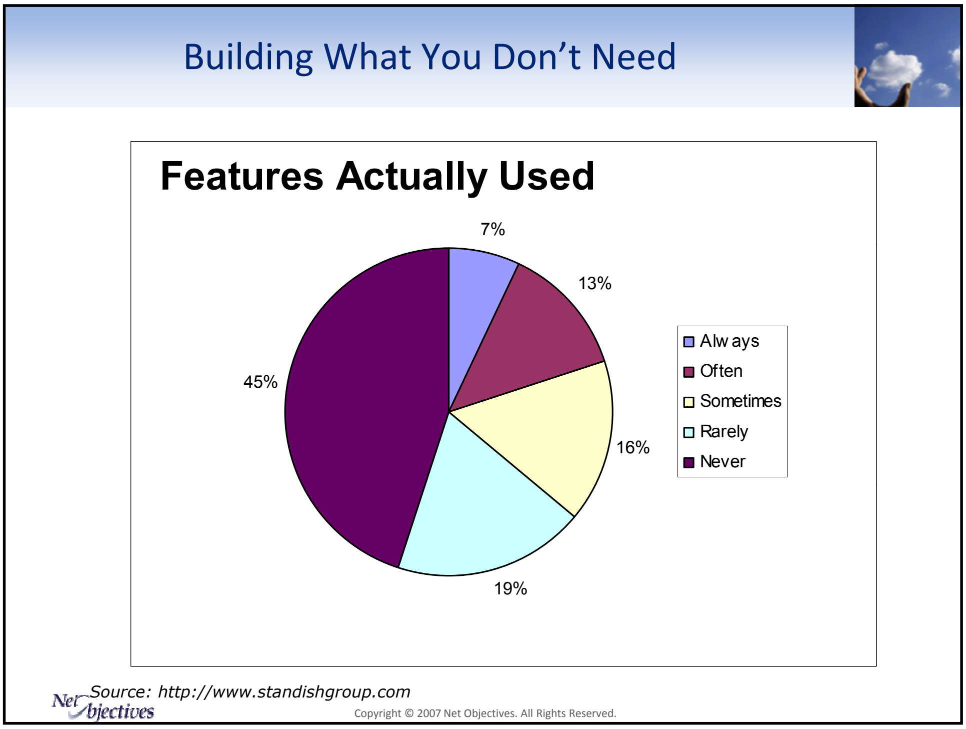

The Addison-Wesley Signature Series
Implementing Lean Software Development
From Concept to Cash
Mary and Tom Poppendieck
Forewords by Jeff Sutherland and Kent Beck

http://www.poppendieck.com

Biggest Wastes According To Lean

- Building what you don’t need.
- Building what you need with poor quality.
- Building what you need in an inefficient manner.





Building What You Don't Need

Source: *http://www.standishgroup.com*

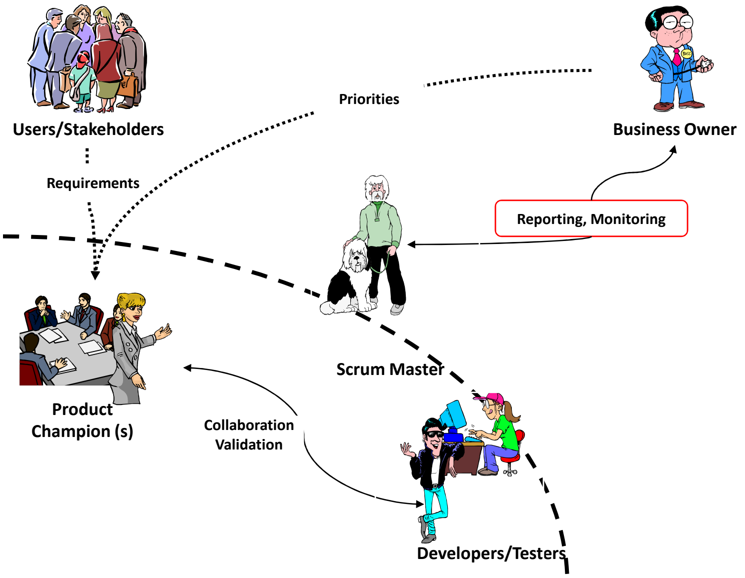
Net Objectives

Net Objectives

info@netobjectives.com
www.netobjectives.com

Scrum Overview

Agile Roles


Users/Stakeholders

Requirements

Priorities

Business Owner

Reporting, Monitoring

Scrum Master

Product Champion (s)

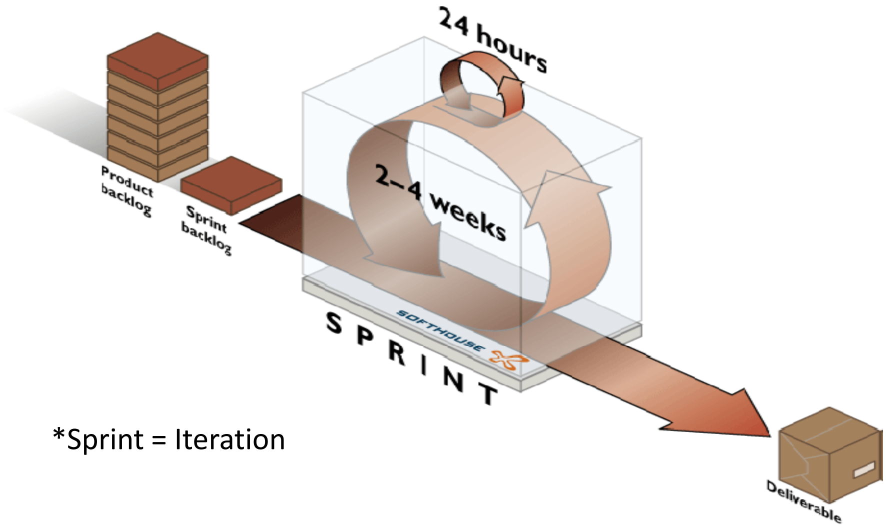
Collaboration Validation

Developers/Testers

What Is Scrum?

- Building software in “sprints” (aka “iterations”)
  - Strict definition is 30 days per sprint
  - Practical is a bit different
    - From 1 to 4 weeks
    - Most teams do 2 weeks
- Product backlog
  - Consisting of stories (typically 1-3 days each)
  - Prioritized by the Product Owner/Champion
- A Cross-Functional Team
  - Members have skills, not roles
  - Scrum Master facilitates team
- *Brief* daily “stand-up” (or “scrum”) meetings

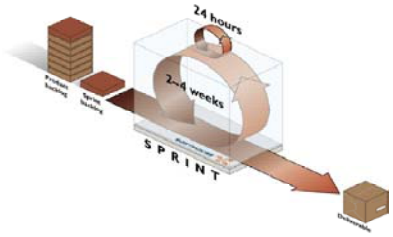
Basic Scrum Flow



*Sprint = Iteration

The Basic Unit Is the Iteration

- For Each Iteration you:
  - Plan
    - What do we need to do?
    - The team decides how much it can do
  - Perform
    - Demonstrable Business Value
    - Potentially shippable Product
    - Quality Process
    - Daily Monitoring
  - Evaluate
    - The Product
    - The Process
- Repeat…

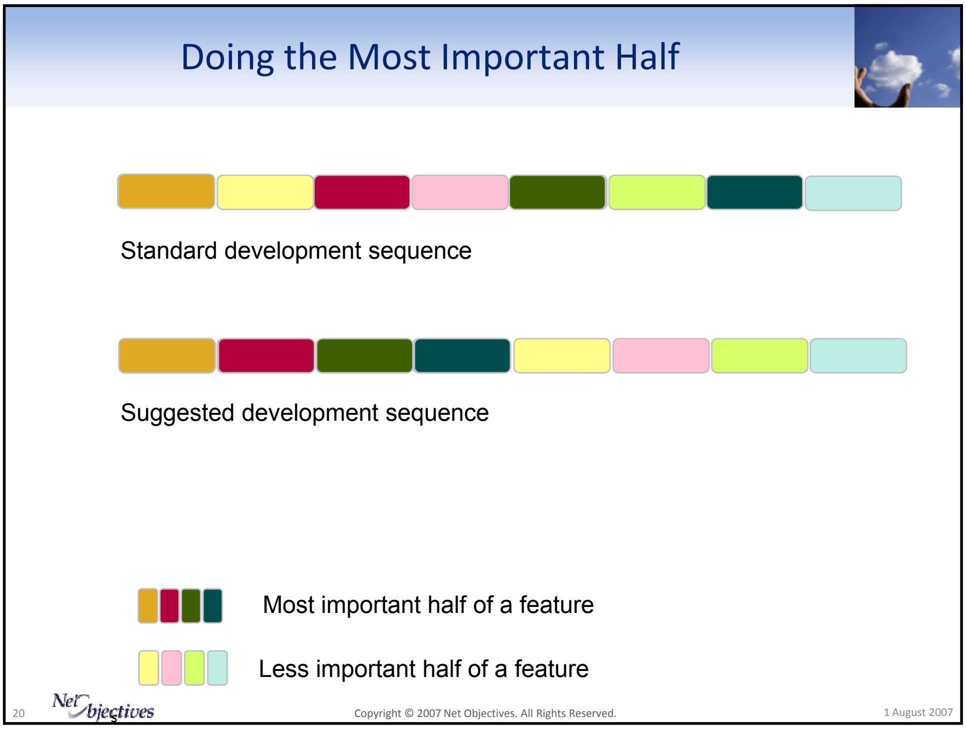


info@netobjectives.com
www.netobjectives.com

The Idea Behind Iterating

Play to Win…

…this hand, and the next!



Doing the Most Important Half

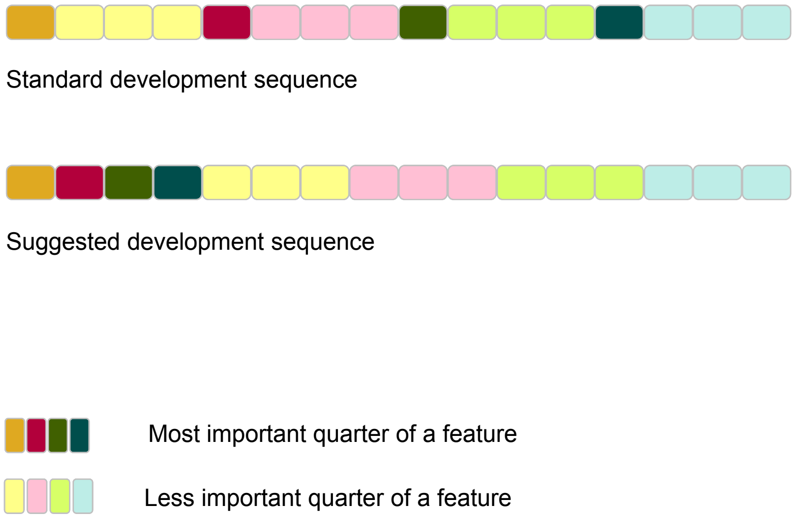
Standard development sequence

Suggested development sequence

Most important half of a feature

Less important half of a feature

Doing the Most Important 25%



Standard development sequence

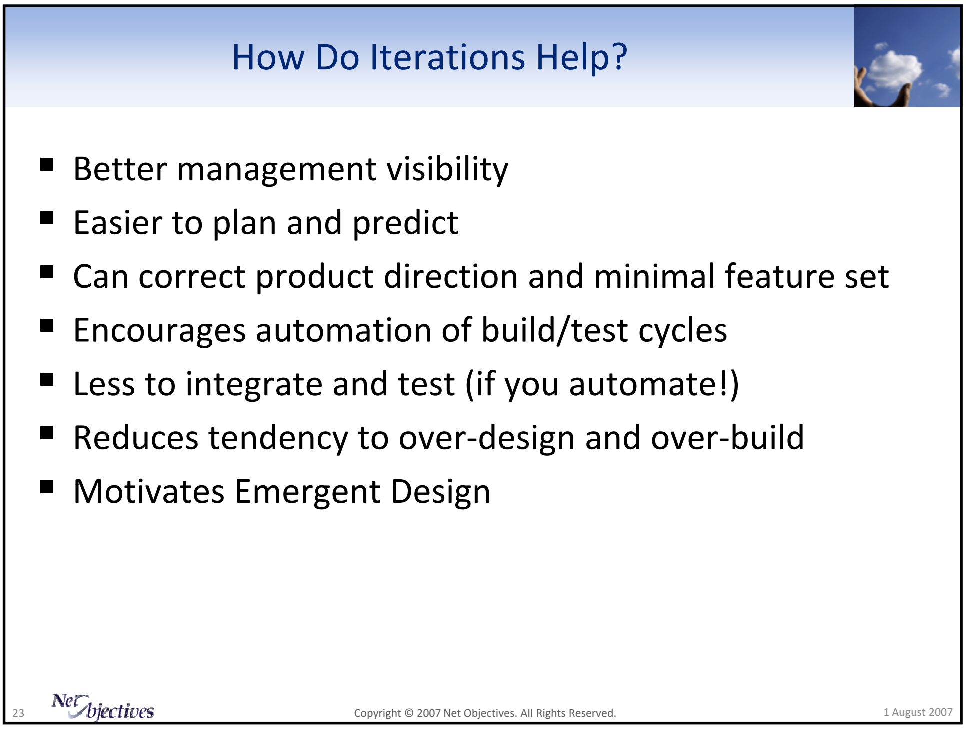
Suggested development sequence

Most important quarter of a feature

Less important quarter of a feature

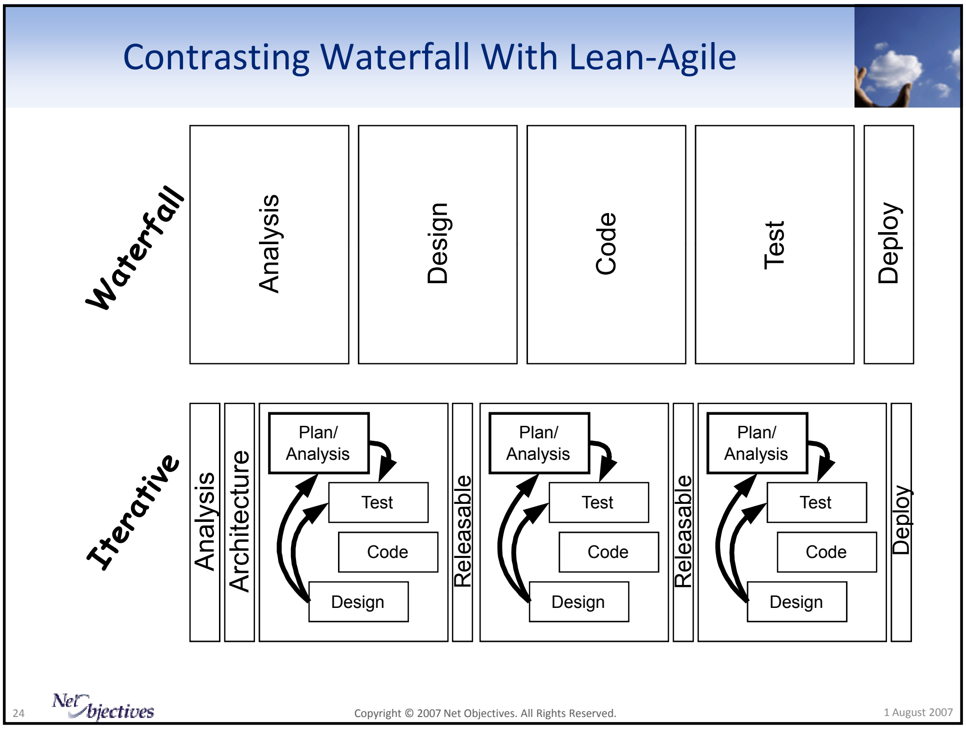
"Deliver" System in Stages

- Concentrate on the known, valuable, features
- Product Champions are more certain about the most valuable features. Benefits:
  - Gives value sooner
  - Creates clarity for what's next
  - Development team gains knowledge as they go
- Lowers risk
  - Of building what you don't need
  - Of overbuilding what you do need
  - Of running out of time or money *too soon*

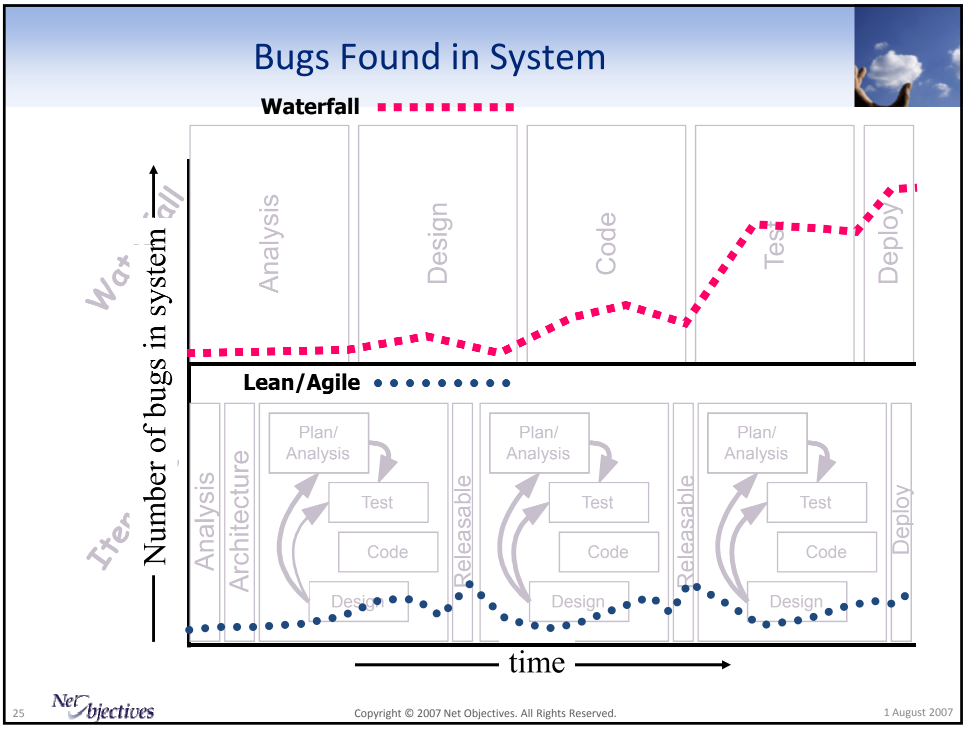


How Do Iterations Help?

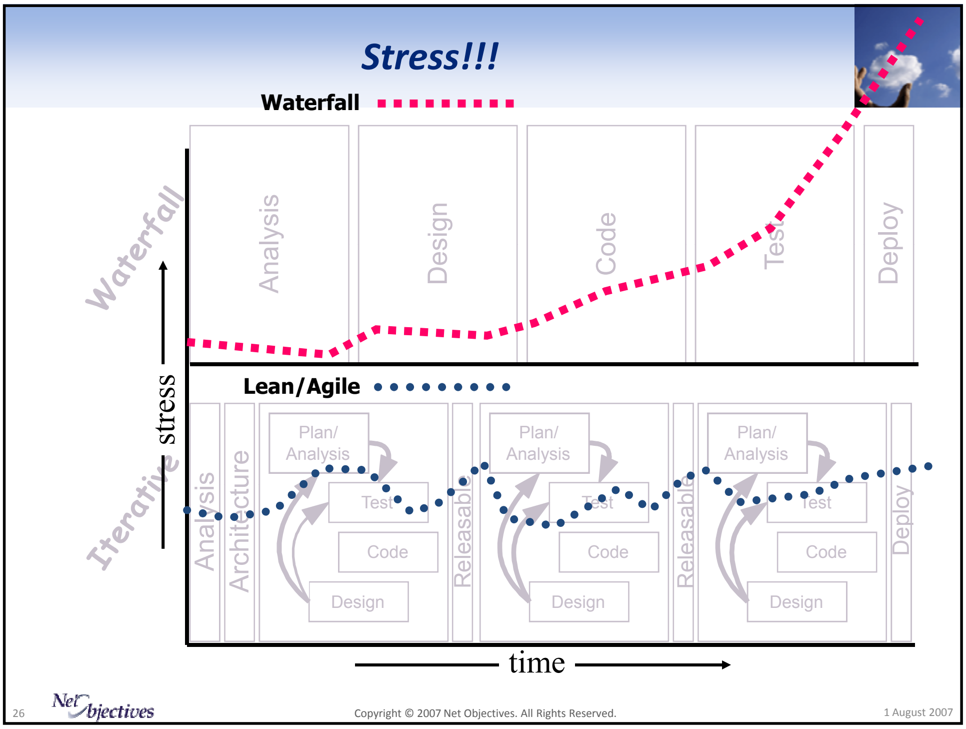
- Better management visibility
- Easier to plan and predict
- Can correct product direction and minimal feature set
- Encourages automation of build/test cycles
- Less to integrate and test (if you automate!)
- Reduces tendency to over-design and over-build
- Motivates Emergent Design



Contrasting Waterfall With Lean-Agile



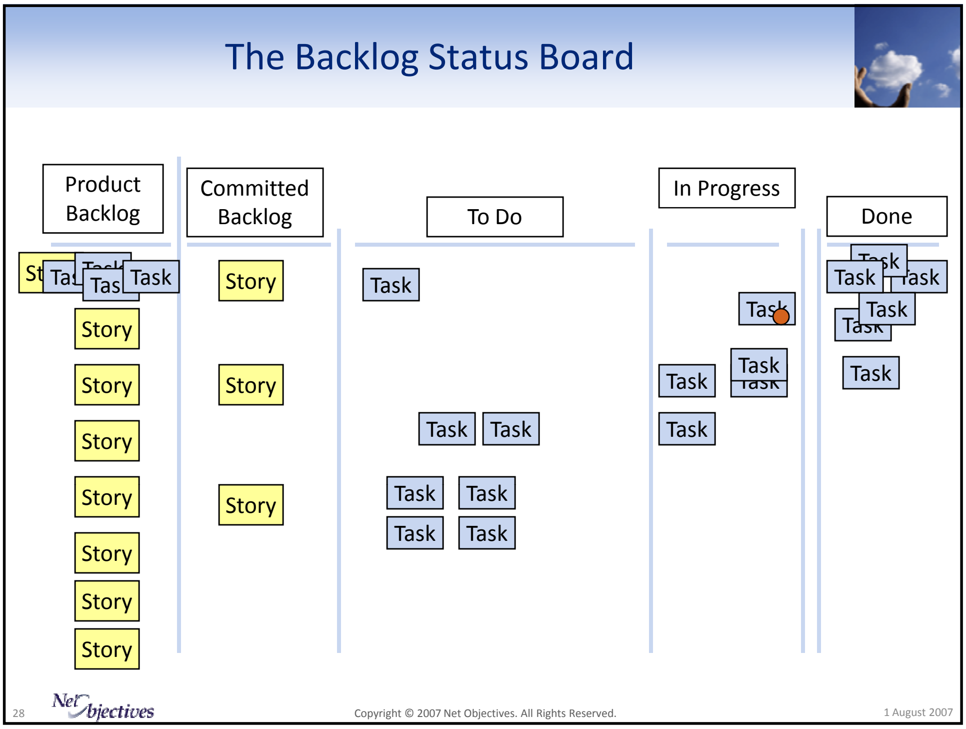
Bugs Found in System



Stress!!!

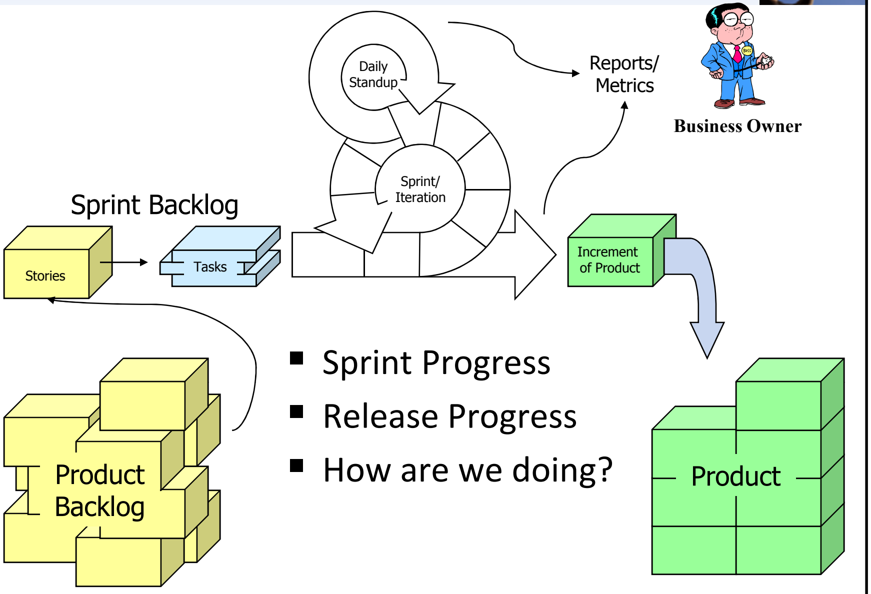
info@netobjectives.com
www.netobjectives.com

Metrics and "Velocity"



The Backlog Status Board

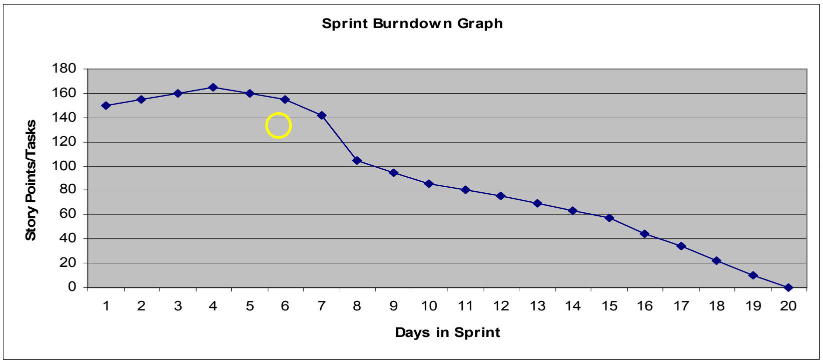
Reporting and Metrics

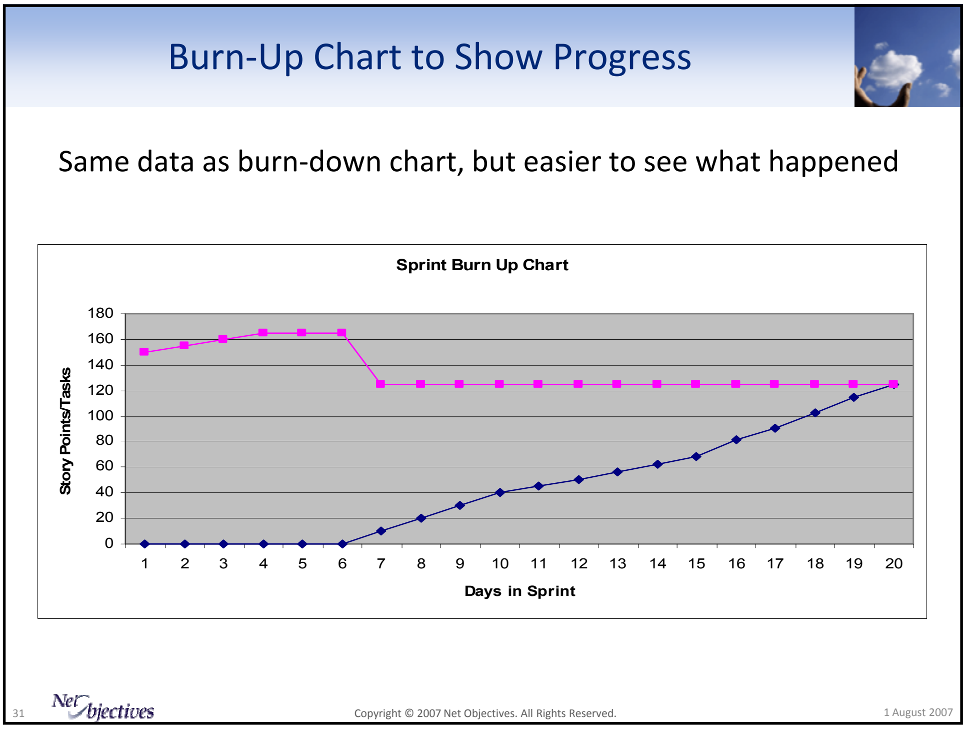


- Sprint Progress
- Release Progress
- How are we doing?

Burn-Down Chart to Show Progress

- Every day throughout the iteration:
  - See how many stories you’ve completed
  - Add up the story-points completed
  - Does the slope of the graph get you to zero at the end?
    - If not, do a midcourse correction (note removed story on day 7)





Burn-Up Chart to Show Progress

Same data as burn-down chart, but easier to see what happened

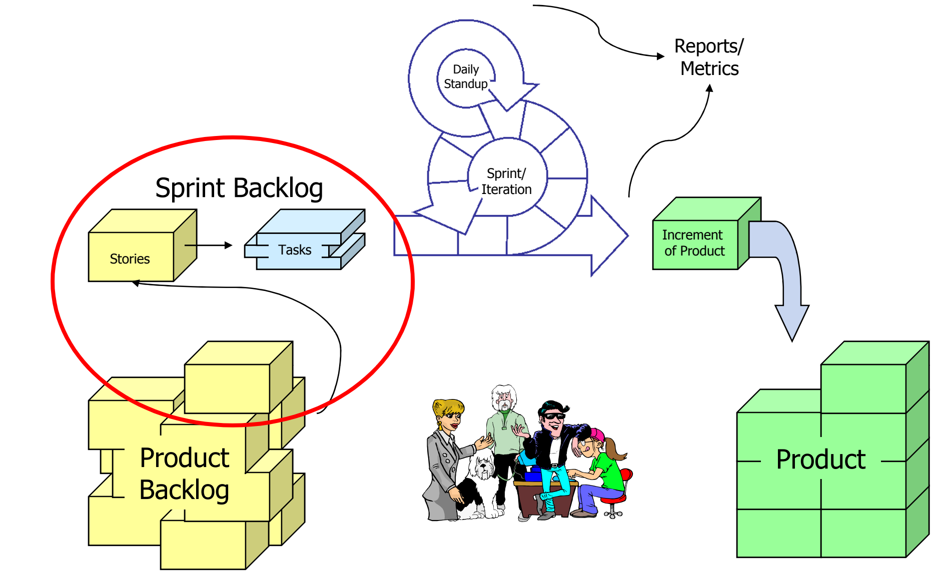
Net Objectives

Net Objectives

info@netobjectives.com
www.netobjectives.com

Scrum Meetings

Sprint Planning Meeting



Sprint Planning Meeting Rules

1 day

1st - 3 hours max. for DevTeam and Product Owner to set sprint goals and select subset of Product Backlog to work on (PO job – prioritize work todo)

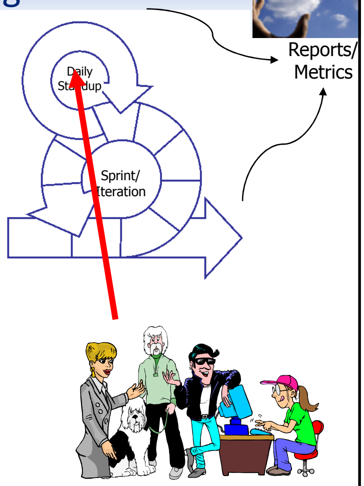
2nd - 3 hours max. for DevTeam and OnsiteCustomers to define Sprint Backlog to build functionality (decompose to tasks and provide estimates)

3rd – 1 hour max. Get back together and decide what we will actually do

Arrr! They be more what you'd call guidelines than actual rules.

Daily Stand-Up Meeting

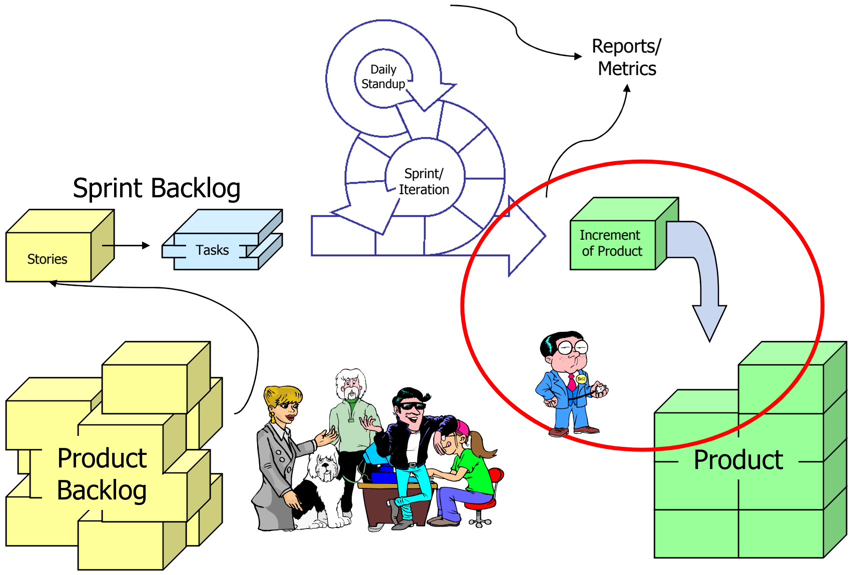
- 15 minutes every day
- Yep…standing (if physically possible)
- Three(+) questions:
  - What did you do yesterday?
  - What will you do today?
  - What is standing in your way?
  - + how many hours left on your current task?
  - + how are you feeling today (1-5)?
- Think about "How is this valuable to the team?"



Scrum Master Visibly Documents All Impediments

- Examples:
  - The test machine is not ready
  - The build process requires too much "hand-holding"
  - Joe Stakeholder is not available to validate a Story
  - Sally getting torn away for bug-fixing on the current release

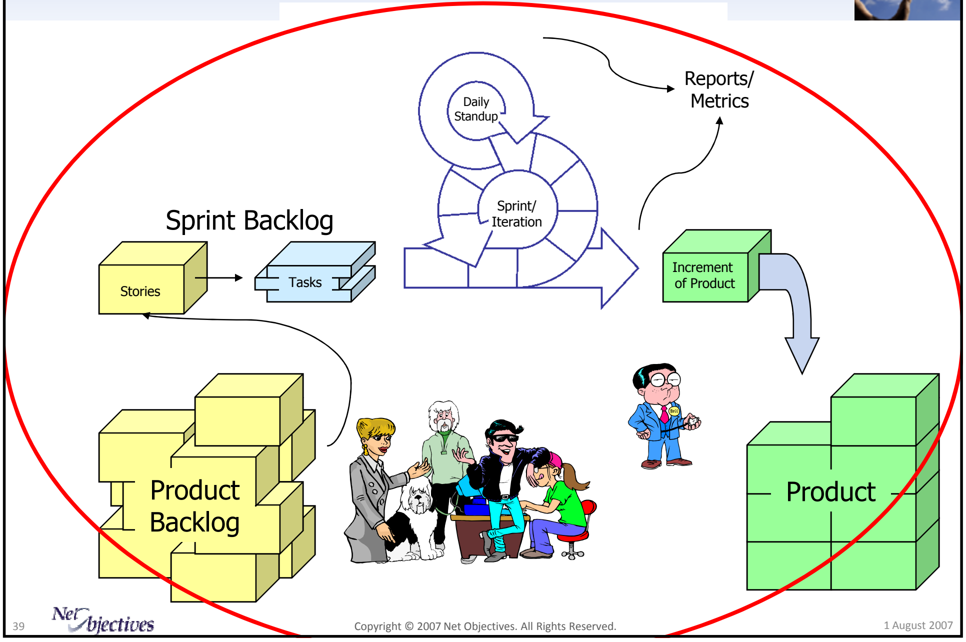
Sprint Review



Sprint Review

- The Team presents Sprint results to Product Champion and other Stakeholders
  - Demonstrate the Product
  - Present results of Functional Tests
  - Describe progress towards release
  - Raise significant, unresolved impediments
  - Show anything else you want to communicate
- Note – ***This is an informal presentation!***
  - No powerpoint necessary, just show off the new features
  - Purpose is to provide visibility, and *inspiration*

Retrospective

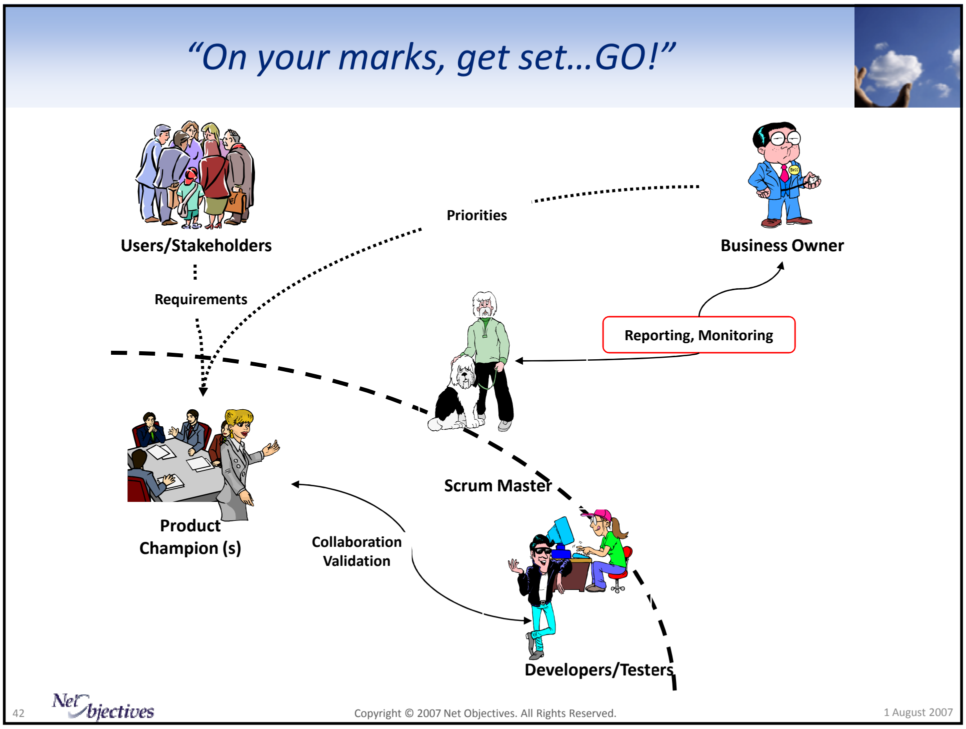


Retrospective

- Whole team is present.
- Usually a small set of questions:
  - What went well?
  - What would we like to change?
  - How can we implement that change?
- Note how these questions are phrased. This is not a whining session, nor is it BLAMEstorming.
- Write answers on the whiteboard as they arise (not into a spreadsheet).
- Often done immediately after the Review, or immediately before the next planning meeting, but you should provide a significant, mandatory break between the two meetings.
- Should be more of a celebration than a wake.
  - But you can't fake it. Be honest and realistic.
  - Order food! There is a reason why food is served at both weddings and wakes.

info@netobjectives.com
www.netobjectives.com

Putting it all Together



"On your marks, get set…GO!"

Users/Stakeholders

Requirements

Priorities

Business Owner

Reporting, Monitoring

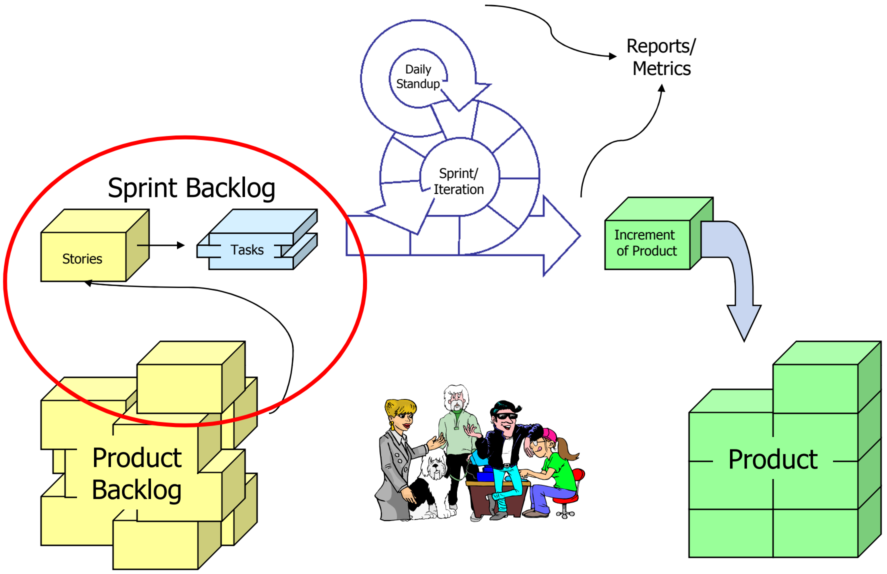
Scrum Master

Product Champion (s)

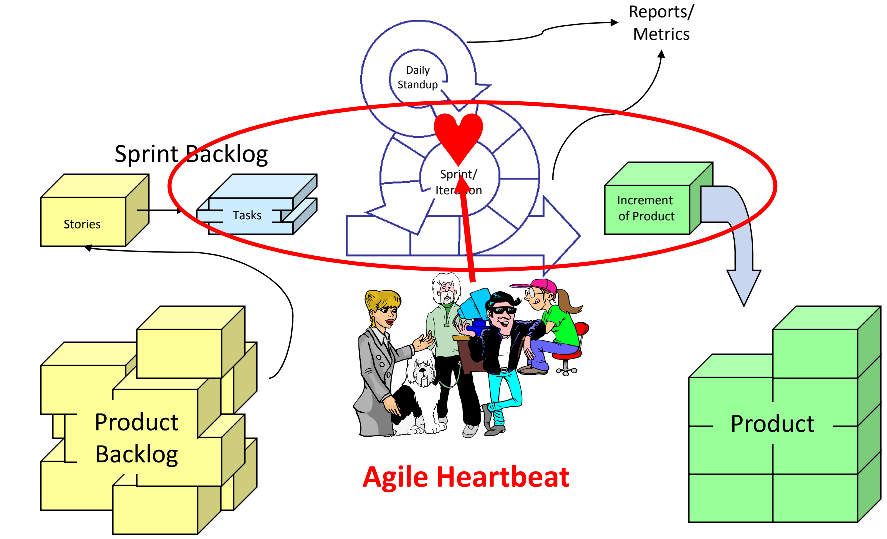
Collaboration
Validation

Developers/Testers

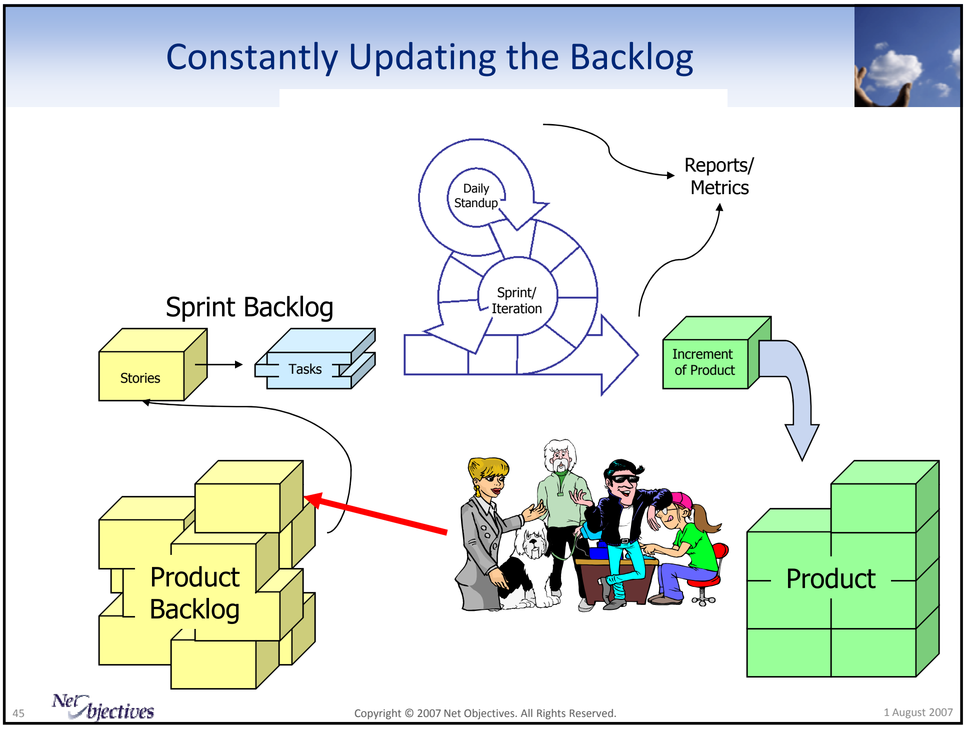
Sprint Planning Meeting



Doing the Work



Agile Heartbeat



Constantly Updating the Backlog

Daily Standup

Sprint/ Iteration

Reports/ Metrics

Sprint Backlog

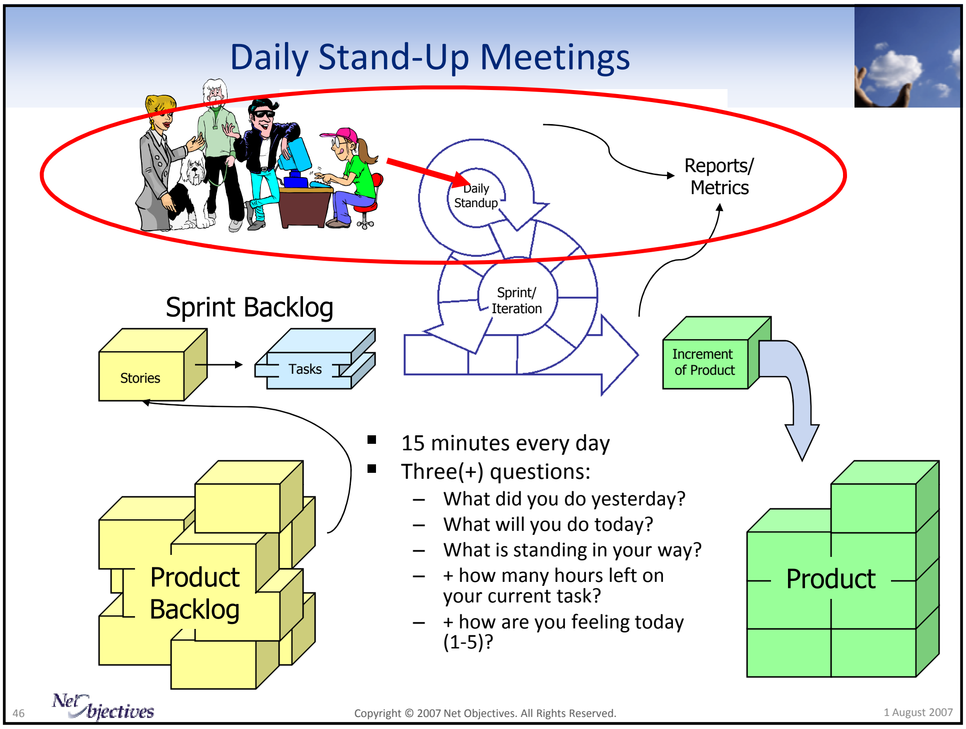
Stories

Tasks

Increment of Product

Product Backlog

Product



Daily Stand-Up Meetings

Daily Standup

Sprint/ Iteration

Reports/ Metrics

Sprint Backlog

Stories

Tasks

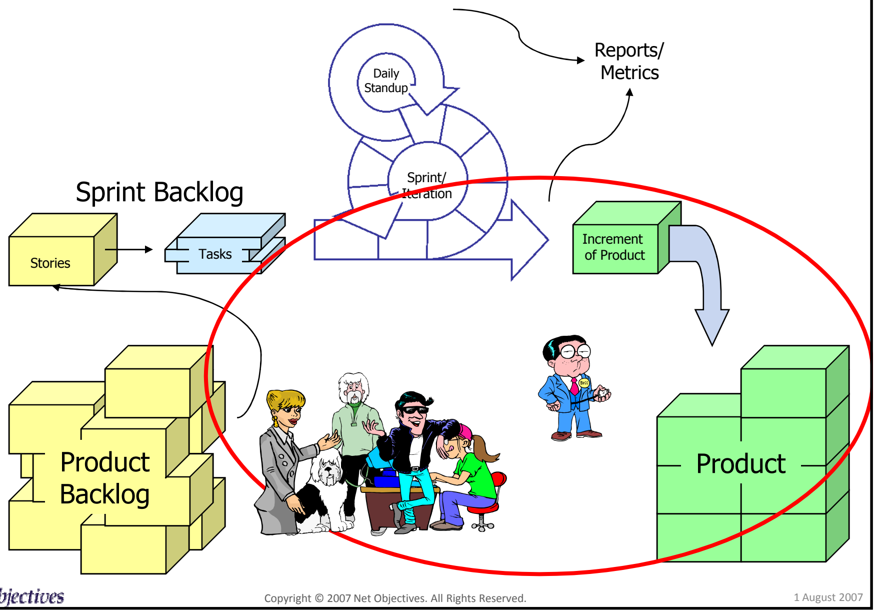
Increment of Product

Product Backlog

Product

- 15 minutes every day
- Three(+) questions:
  - What did you do yesterday?
  - What will you do today?
  - What is standing in your way?
  - + how many hours left on your current task?
  - + how are you feeling today (1-5)?

Review and Retrospective

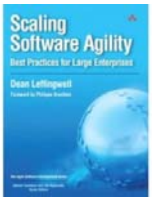


info@netobjectives.com
www.netobjectives.com

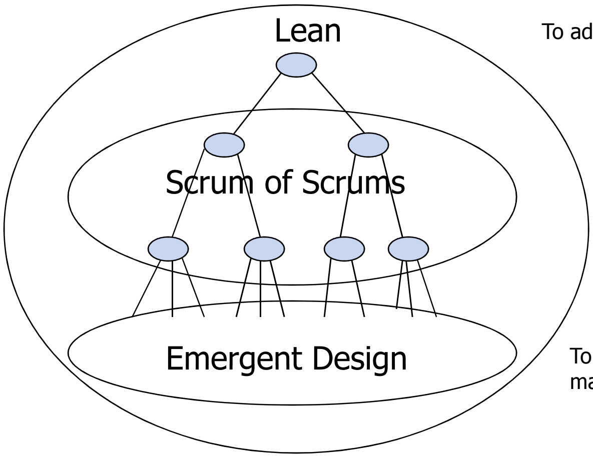
Scaling Scrum

Agile Enterprises

- Intentional architecture
- Lean requirements at scale: vision, roadmap and just-in-time elaboration
- Systems of systems and the Agile release train
- Managing highly distributed teams
- Impact on customers and operations
- Changing the organization
- Measuring business performance



Lean-Scrum-Emergent Design

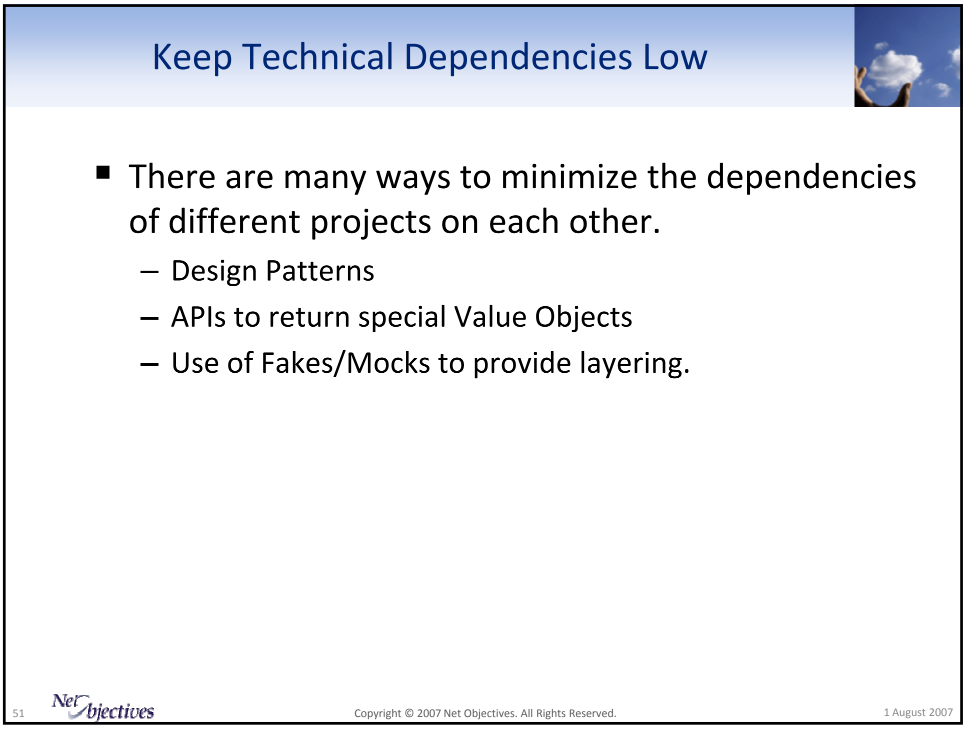


To address structure of the organization

To coordinate teams

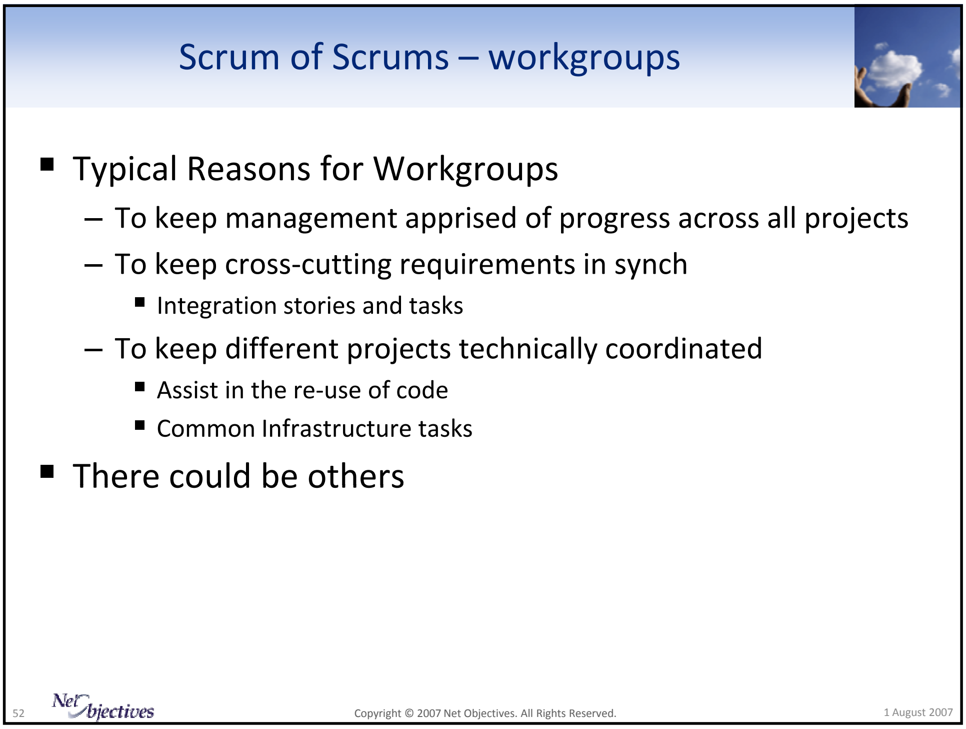
To alter code while maintaining quality

Scrum does scale, but it needs more than knowledge and practices of Scrum to scale successfully.



Keep Technical Dependencies Low

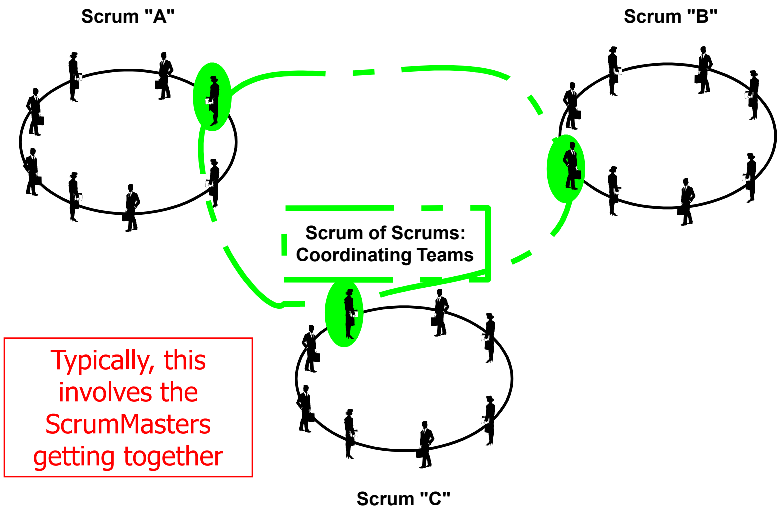
- There are many ways to minimize the dependencies of different projects on each other.
  - Design Patterns
  - APIs to return special Value Objects
  - Use of Fakes/Mocks to provide layering.



Scrum of Scrums – workgroups

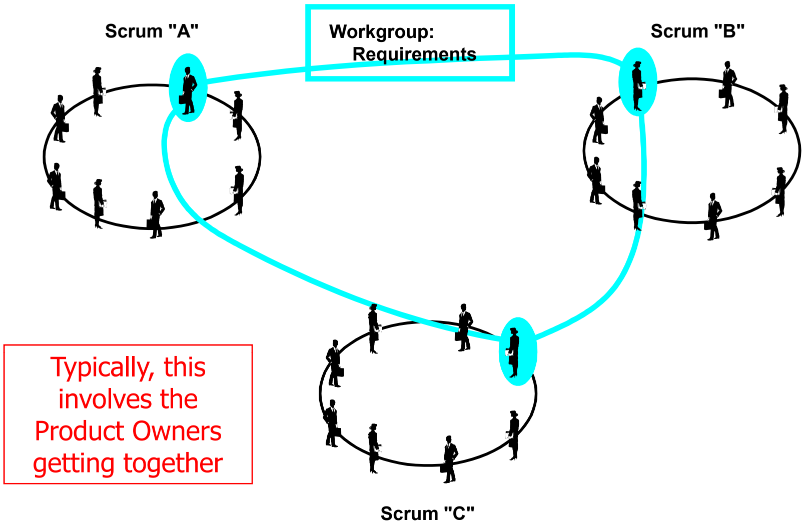
- Typical Reasons for Workgroups
  - To keep management apprised of progress across all projects
  - To keep cross-cutting requirements in synch
    - Integration stories and tasks
  - To keep different projects technically coordinated
    - Assist in the re-use of code
    - Common Infrastructure tasks
- There could be others

Scrum of Scrums – Coordinating Teams and Reporting to Management



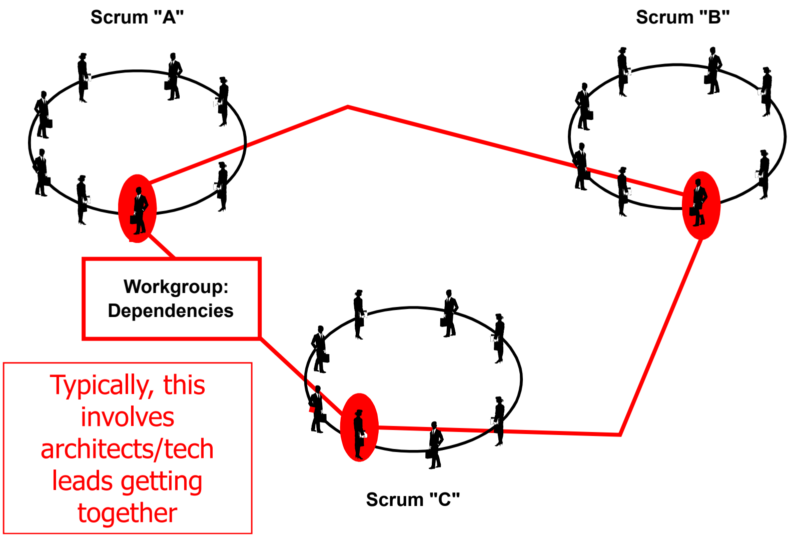
Typically, this involves the ScrumMasters getting together

Workgroup: Cross-Cutting Requirements



Typically, this involves the Product Owners getting together

Workgroup: Technical Dependencies

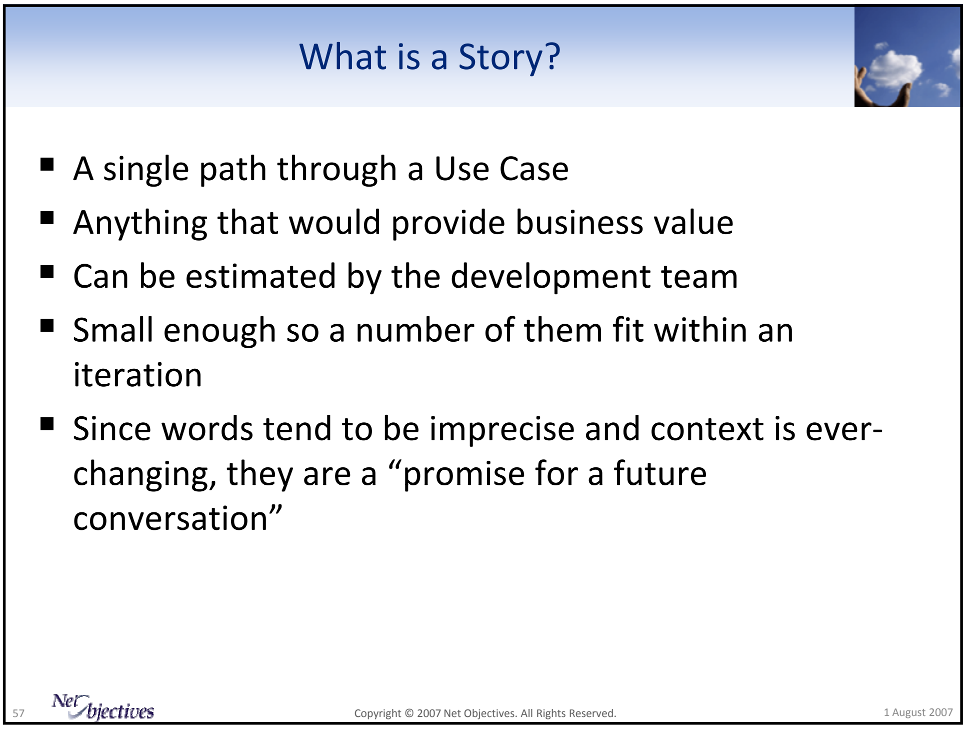


Typically, this involves architects/tech leads getting together

Net Objectives

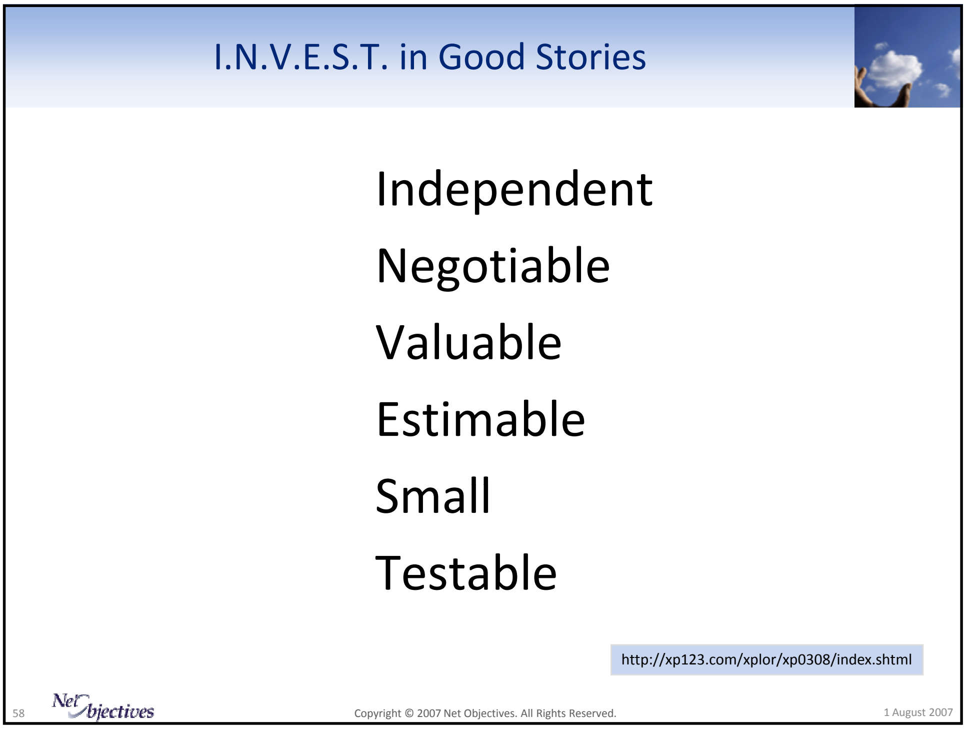
info@netobjectives.com
www.netobjectives.com

The Story Behind Stories



What is a Story?

- A single path through a Use Case
- Anything that would provide business value
- Can be estimated by the development team
- Small enough so a number of them fit within an iteration
- Since words tend to be imprecise and context is ever-changing, they are a “promise for a future conversation”



I.N.V.E.S.T. in Good Stories

Independent

Negotiable

Valuable

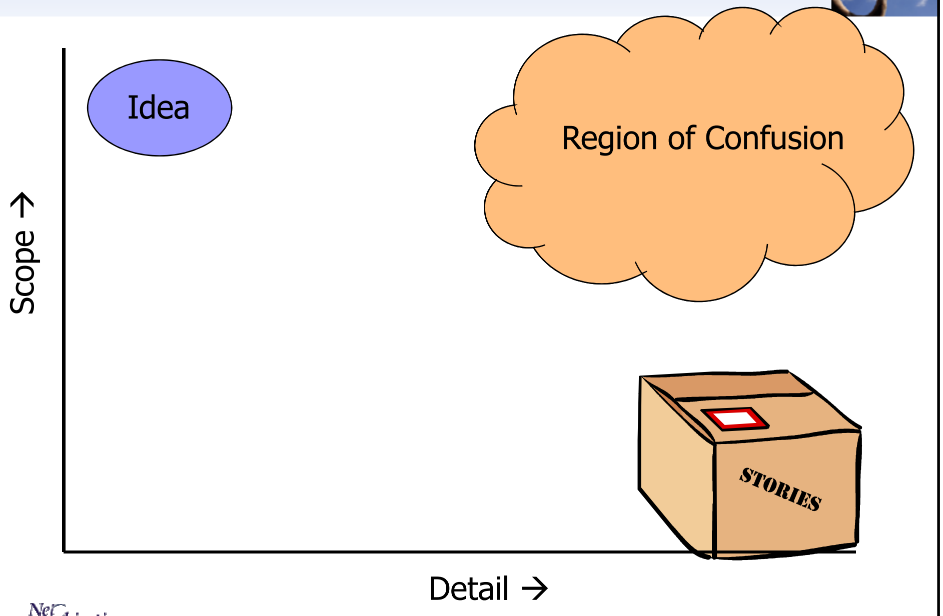
Estimable

Small

Testable

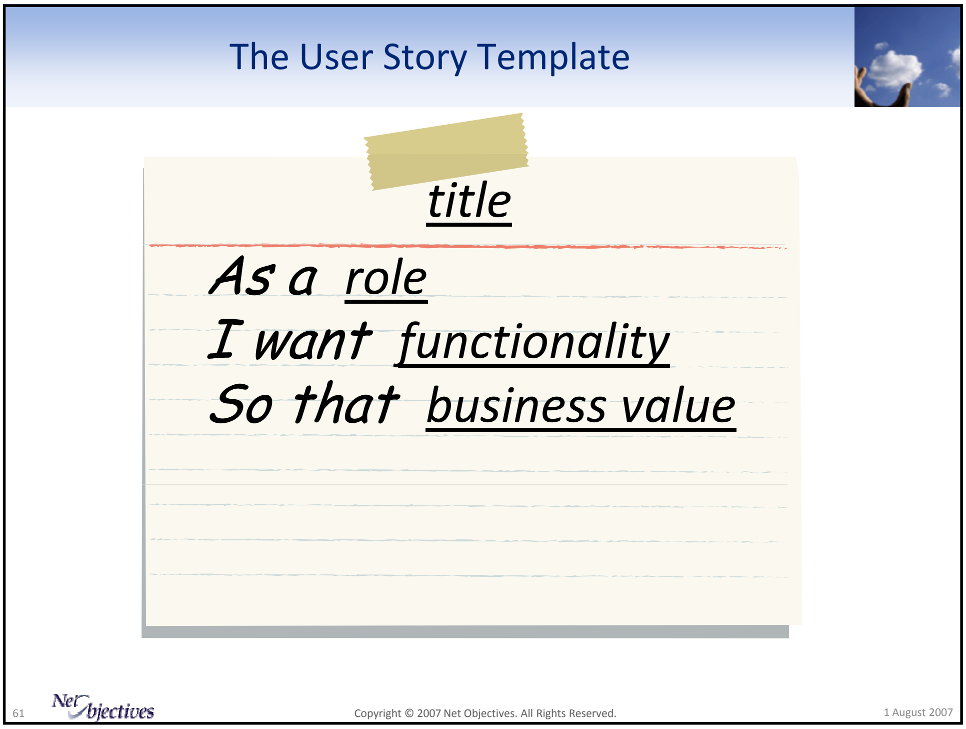
http://xp123.com/xplor/xp0308/index.shtml

Unfolding Stories



User Roles/Personas

- Multiple personas provide a richer view and better stories
- Vary Users by
  - What they want to accomplish
  - How often the software will be used
  - How they use the software
  - Domain expertise
  - Computer proficiency
  - Product proficiency
- Software solves the needs of real people
- You can converse about the persona



The User Story Template

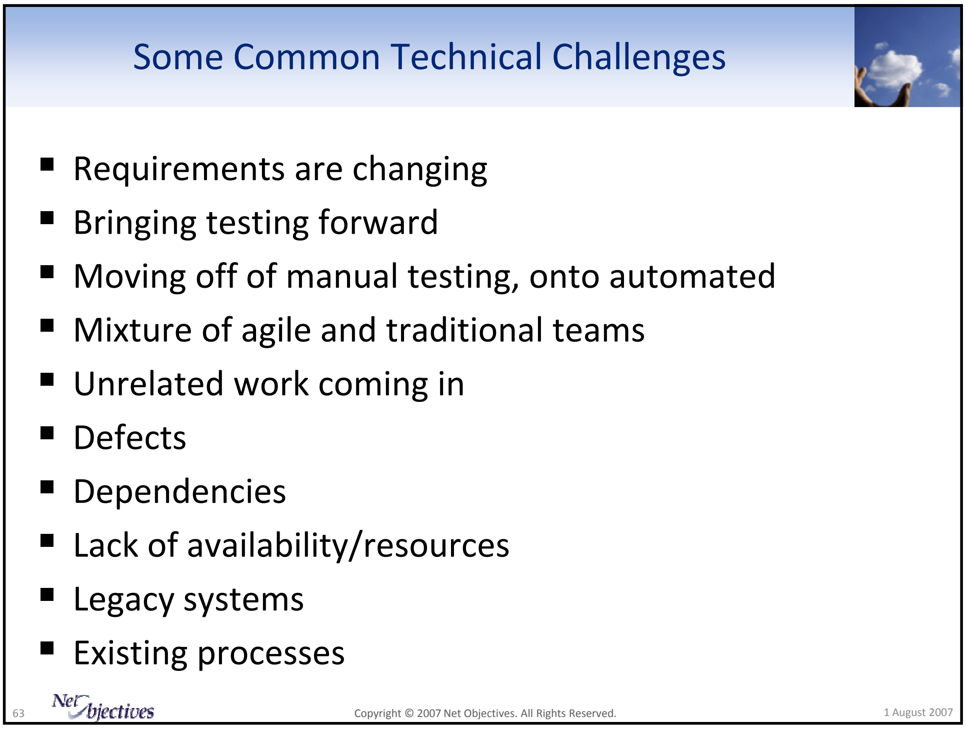
title

As a role
I want functionality
So that business value

info@netobjectives.com
www.netobjectives.com

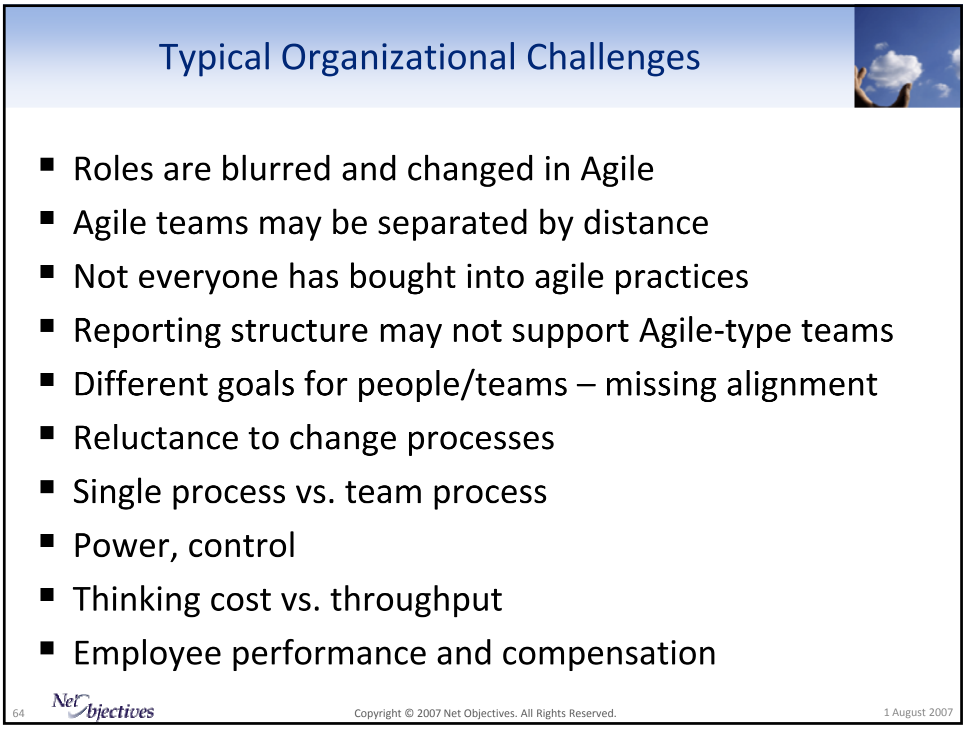
Challenges, Challenges

No one said it would be easy.



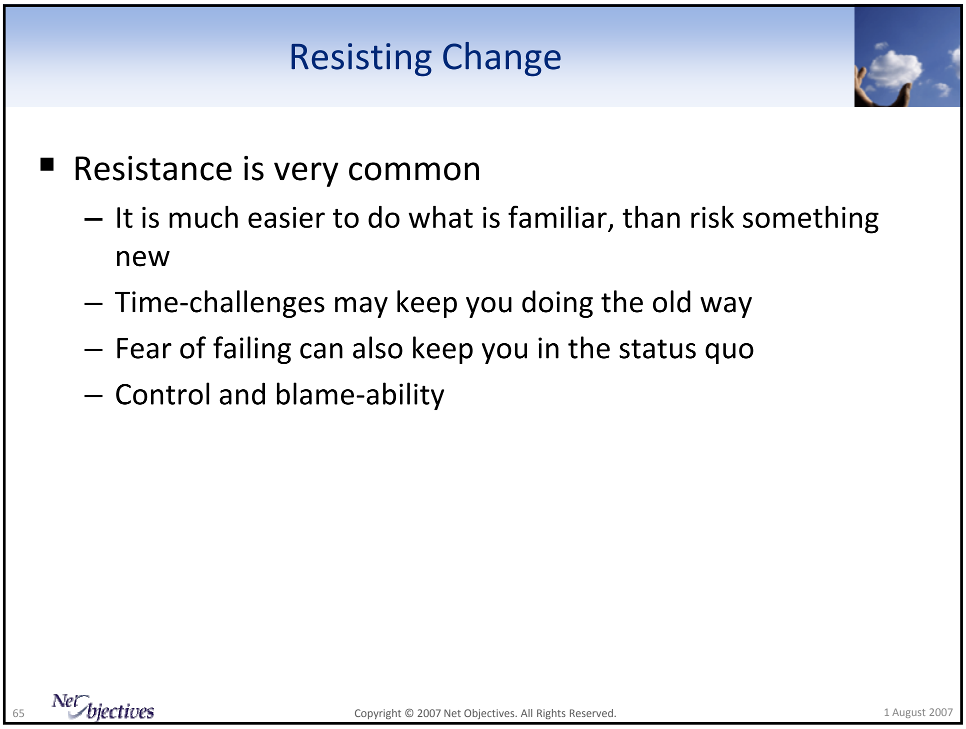
Some Common Technical Challenges

- Requirements are changing
- Bringing testing forward
- Moving off of manual testing, onto automated
- Mixture of agile and traditional teams
- Unrelated work coming in
- Defects
- Dependencies
- Lack of availability/resources
- Legacy systems
- Existing processes



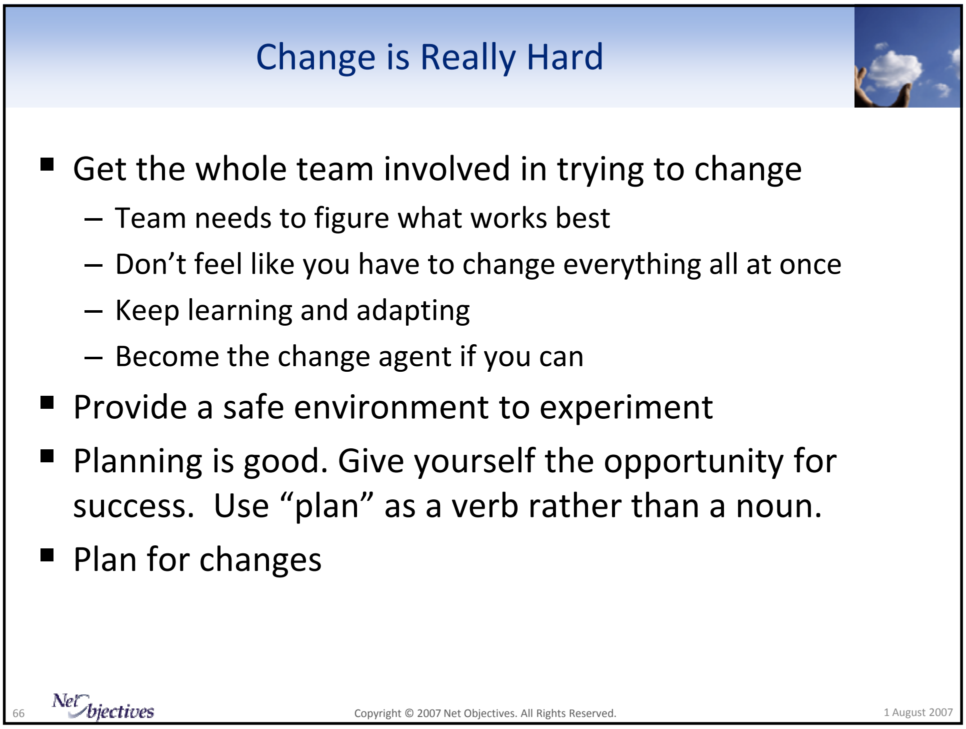
Typical Organizational Challenges

- Roles are blurred and changed in Agile
- Agile teams may be separated by distance
- Not everyone has bought into agile practices
- Reporting structure may not support Agile-type teams
- Different goals for people/teams – missing alignment
- Reluctance to change processes
- Single process vs. team process
- Power, control
- Thinking cost vs. throughput
- Employee performance and compensation



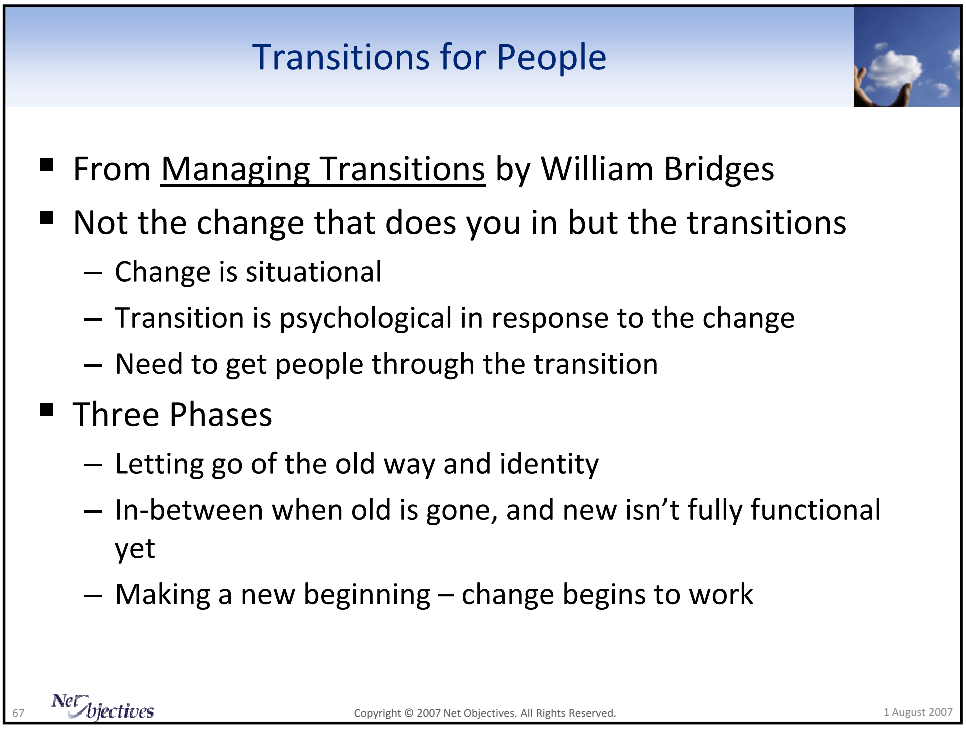
Resisting Change

- Resistance is very common
  - It is much easier to do what is familiar, than risk something new
  - Time-challenges may keep you doing the old way
  - Fear of failing can also keep you in the status quo
  - Control and blame-ability



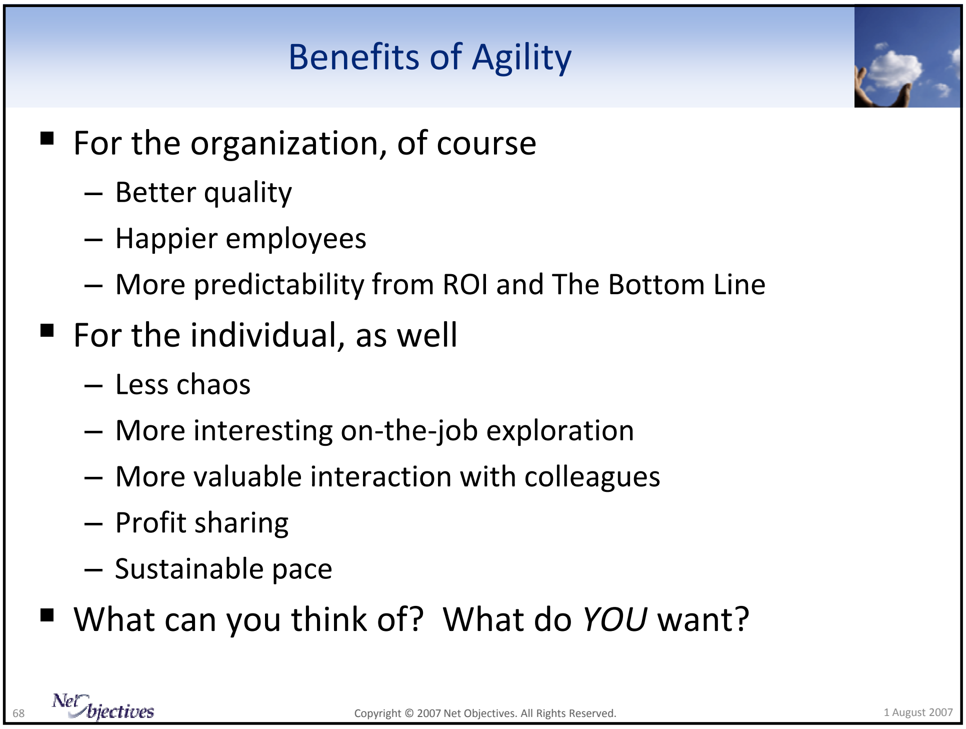
Change is Really Hard

- Get the whole team involved in trying to change
  - Team needs to figure what works best
  - Don't feel like you have to change everything all at once
  - Keep learning and adapting
  - Become the change agent if you can
- Provide a safe environment to experiment
- Planning is good. Give yourself the opportunity for success. Use "plan" as a verb rather than a noun.
- Plan for changes



Transitions for People

- From Managing Transitions by William Bridges
- Not the change that does you in but the transitions
  - Change is situational
  - Transition is psychological in response to the change
  - Need to get people through the transition
- Three Phases
  - Letting go of the old way and identity
  - In-between when old is gone, and new isn't fully functional yet
  - Making a new beginning – change begins to work



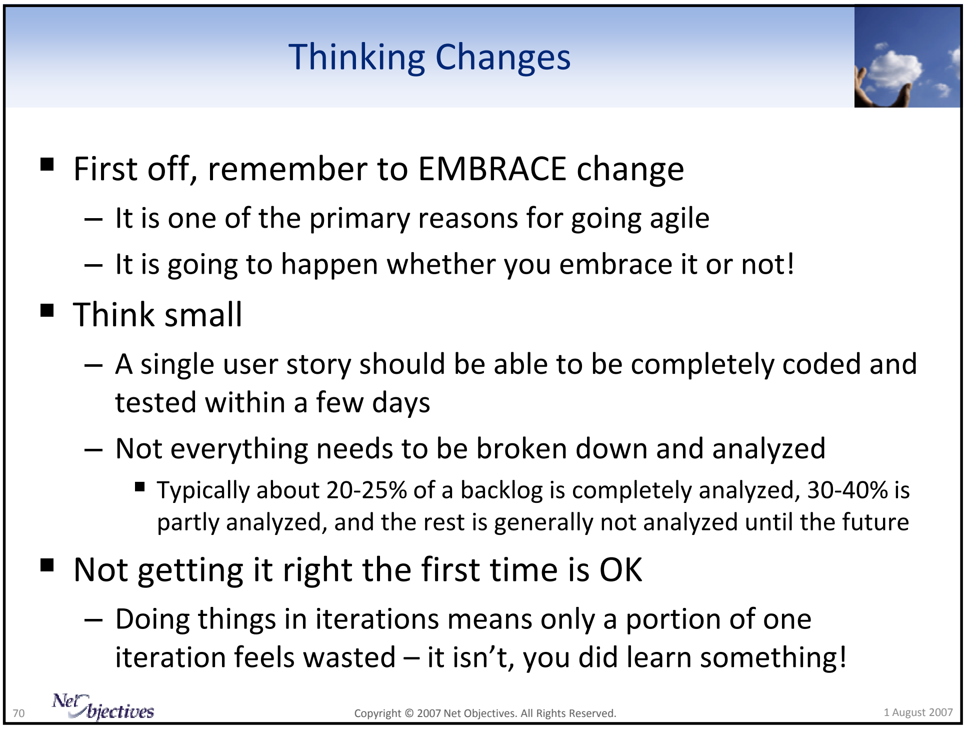
Benefits of Agility

- For the organization, of course
  - Better quality
  - Happier employees
  - More predictability from ROI and The Bottom Line
- For the individual, as well
  - Less chaos
  - More interesting on-the-job exploration
  - More valuable interaction with colleagues
  - Profit sharing
  - Sustainable pace
- What can you think of? What do *YOU* want?

info@netobjectives.com
www.netobjectives.com

Summary

Take-aways for Business Analysts



Thinking Changes

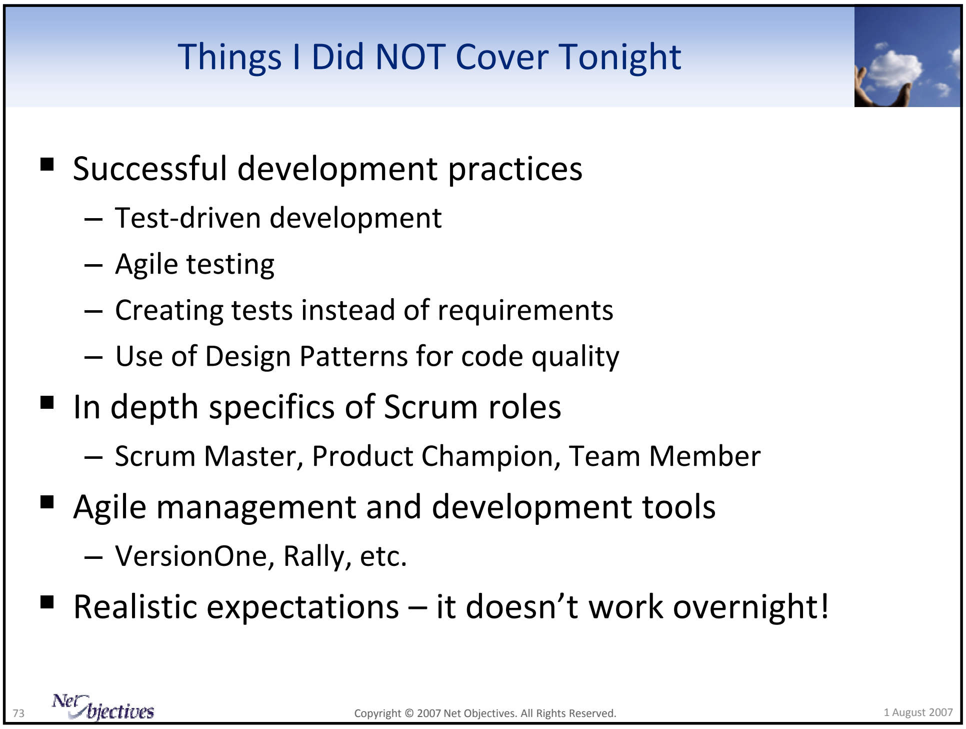
- First off, remember to EMBRACE change
  - It is one of the primary reasons for going agile
  - It is going to happen whether you embrace it or not!
- Think small
  - A single user story should be able to be completely coded and tested within a few days
  - Not everything needs to be broken down and analyzed
    - Typically about 20-25% of a backlog is completely analyzed, 30-40% is partly analyzed, and the rest is generally not analyzed until the future
- Not getting it right the first time is OK
  - Doing things in iterations means only a portion of one iteration feels wasted – it isn’t, you did learn something!

Operational Changes

- What you do on a daily basis is slightly different (or a lot different depending on your role!)
  - Remember, a story is an invitation to a future conversation
    - You will have LOTS of communication with developers
    - You will have LOTS of communication with testers
  - You need to build "analysis runway"
    - The most important items need to be analyzed 1-2 iterations into the future, not YEARS into the future
    - Other roles use your analysis runway to keep iterating
- If you are also the Product Champion, you need to be constantly prioritizing stories
  - Everything has an expectation and a price
  - When the price exceeds the expectation, it should not be built!
- Your goal needs to be to get rid of the 64% of nearly useless features!

It sounds easy, now what?

- Don't kid yourself
  - This is NOT an easy transition to make
  - Business analysts and testers are the positions most impacted by going to an agile process
  - Your communication skills will be tested
  - Can you really take a 30 page requirements document and turn it into a story or stories that fit on index cards?
- Do more research
  - I've only covered the barest essentials to get the point across
  - Most teams that successfully transition to agile use a coach
    - "Been there, done that" helps a lot
    - Keeps the team honest to the goals and principles



Things I Did NOT Cover Tonight

- Successful development practices
  - Test-driven development
  - Agile testing
  - Creating tests instead of requirements
  - Use of Design Patterns for code quality
- In depth specifics of Scrum roles
  - Scrum Master, Product Champion, Team Member
- Agile management and development tools
  - VersionOne, Rally, etc.
- Realistic expectations – it doesn't work overnight!

info@netobjectives.com
www.netobjectives.com

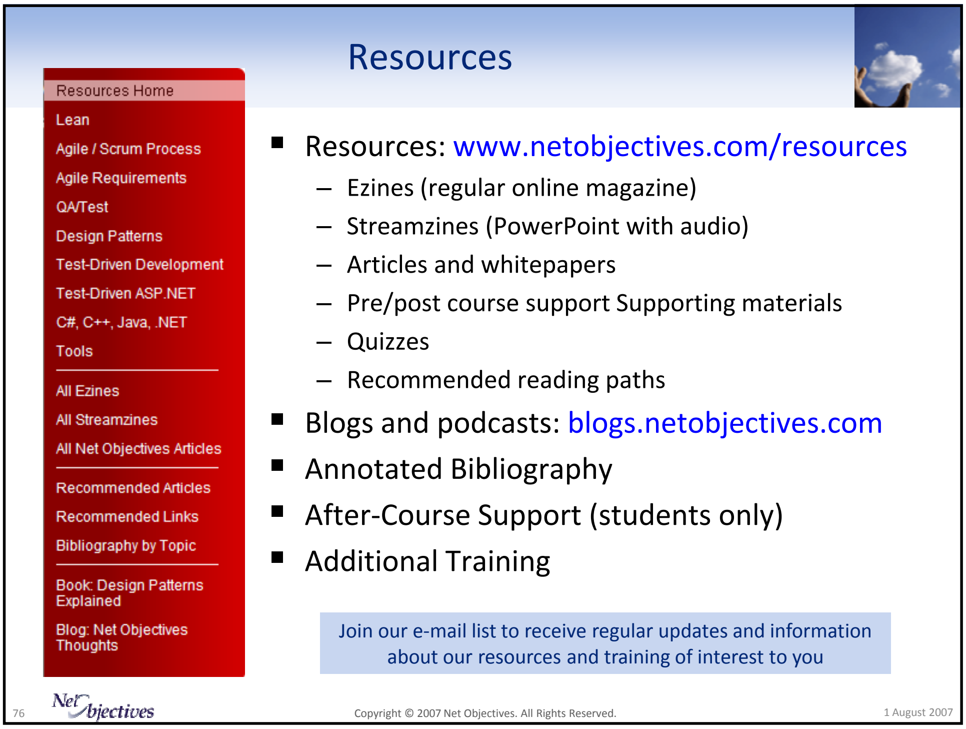
Questions?

Net Objectives

info@netobjectives.com
www.netobjectives.com

Thank You!

… and following is more to help you plan your next steps



Resources

- Resources Home
- Lean
- Agile / Scrum Process
- Agile Requirements
- QA/Test
- Design Patterns
- Test-Driven Development
- Test-Driven ASP.NET
- C#, C++, Java, .NET
- Tools
- All Ezines
- All Streamzines
- All Net Objectives Articles
- Recommended Articles
- Recommended Links
- Bibliography by Topic
- Book: Design Patterns Explained
- Blog: Net Objectives Thoughts

- Resources: www.netobjectives.com/resources
  - Ezines (regular online magazine)
  - Streamzines (PowerPoint with audio)
  - Articles and whitepapers
  - Pre/post course support Supporting materials
  - Quizzes
  - Recommended reading paths
- Blogs and podcasts: blogs.netobjectives.com
- Annotated Bibliography
- After-Course Support (students only)
- Additional Training

Join our e-mail list to receive regular updates and information about our resources and training of interest to you

Upcoming Public Courses

- Bellevue, WA
  - Design Patterns Explained – August 6-8
  - Advanced Software Design – August 9-10
  - Design Patterns Explained – October 22-24
  - Advanced Software Design – October 25-26
- Boulder, CO
  - Lean Quality Assurance – September 12
  - Lean-Agile Testing Practices – September 13-14
- Denver, CO
  - Lean Software Development – October 17-18

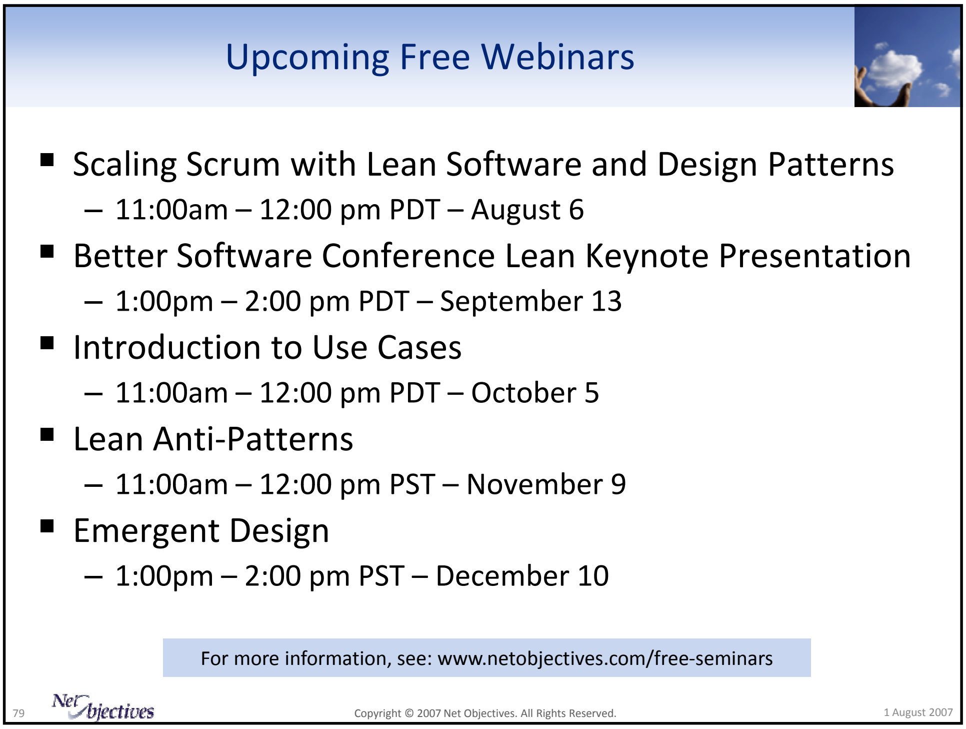
For more information, see www.netobjectives.com/courses

All courses are 9am – 5pm

Upcoming Free Seminars

- Bay Area, CA
  - The Business Value of Pair Programming – August 15

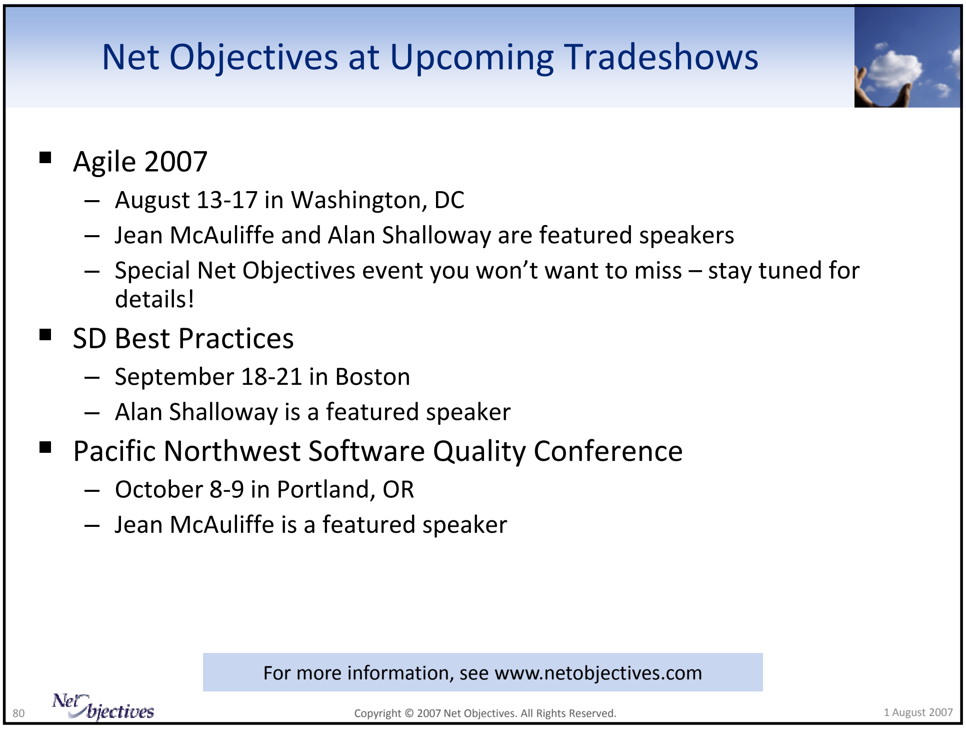
For more information, see: www.netobjectives.com/free-seminars



Upcoming Free Webinars

- Scaling Scrum with Lean Software and Design Patterns
  - 11:00am – 12:00 pm PDT – August 6
- Better Software Conference Lean Keynote Presentation
  - 1:00pm – 2:00 pm PDT – September 13
- Introduction to Use Cases
  - 11:00am – 12:00 pm PDT – October 5
- Lean Anti-Patterns
  - 11:00am – 12:00 pm PST – November 9
- Emergent Design
  - 1:00pm – 2:00 pm PST – December 10

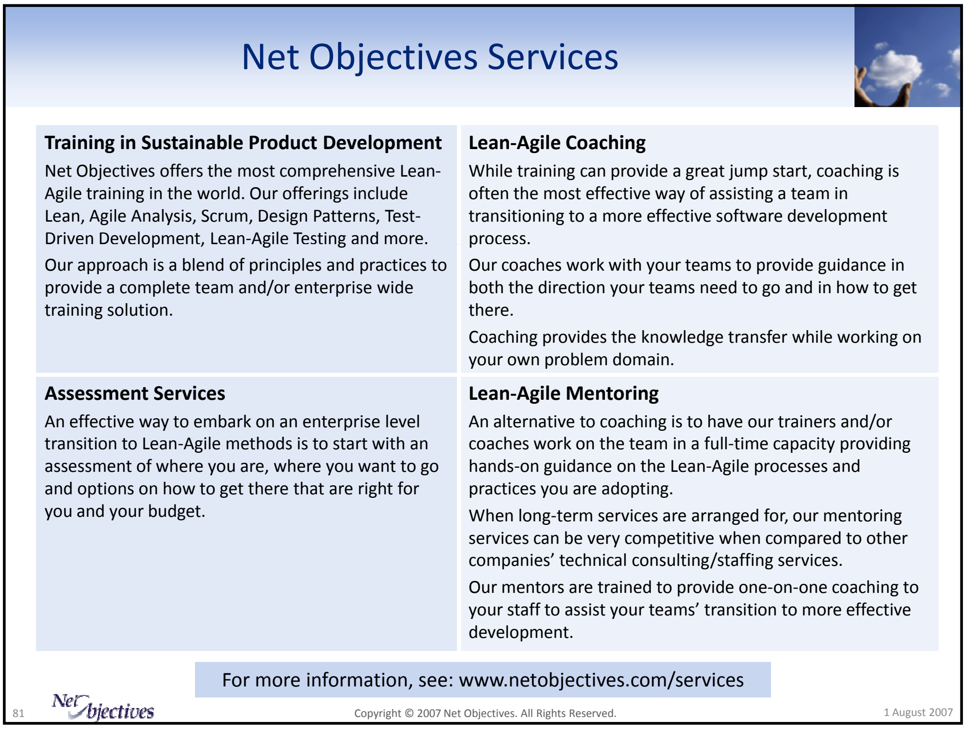
For more information, see: www.netobjectives.com/free-seminars



Net Objectives at Upcoming Tradeshows

- Agile 2007
  - August 13-17 in Washington, DC
  - Jean McAuliffe and Alan Shalloway are featured speakers
  - Special Net Objectives event you won't want to miss – stay tuned for details!
- SD Best Practices
  - September 18-21 in Boston
  - Alan Shalloway is a featured speaker
- Pacific Northwest Software Quality Conference
  - October 8-9 in Portland, OR
  - Jean McAuliffe is a featured speaker

For more information, see www.netobjectives.com



Net Objectives Services

Training in Sustainable Product Development

Net Objectives offers the most comprehensive Lean-Agile training in the world. Our offerings include Lean, Agile Analysis, Scrum, Design Patterns, Test-Driven Development, Lean-Agile Testing and more.

Our approach is a blend of principles and practices to provide a complete team and/or enterprise wide training solution.

Lean-Agile Coaching

While training can provide a great jump start, coaching is often the most effective way of assisting a team in transitioning to a more effective software development process.

Our coaches work with your teams to provide guidance in both the direction your teams need to go and in how to get there.

Coaching provides the knowledge transfer while working on your own problem domain.

Assessment Services

An effective way to embark on an enterprise level transition to Lean-Agile methods is to start with an assessment of where you are, where you want to go and options on how to get there that are right for you and your budget.

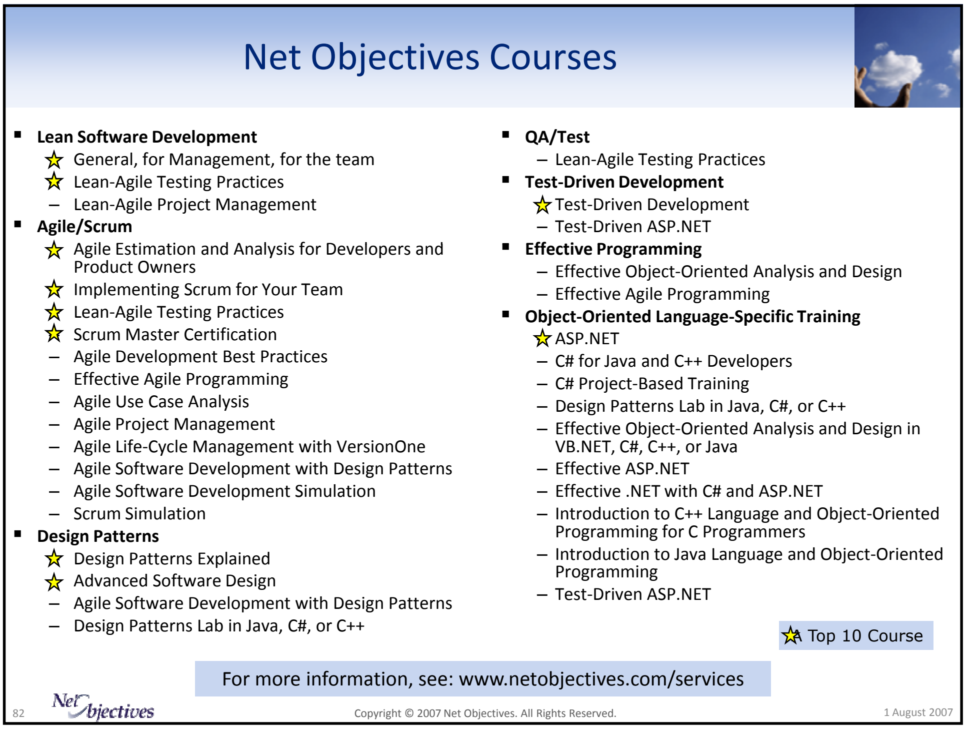
Lean-Agile Mentoring

An alternative to coaching is to have our trainers and/or coaches work on the team in a full-time capacity providing hands-on guidance on the Lean-Agile processes and practices you are adopting.

When long-term services are arranged for, our mentoring services can be very competitive when compared to other companies' technical consulting/staffing services.

Our mentors are trained to provide one-on-one coaching to your staff to assist your teams' transition to more effective development.

For more information, see: www.netobjectives.com/services



Net Objectives Courses

- **Lean Software Development**
  - ☆ General, for Management, for the team
  - ☆ Lean-Agile Testing Practices
  - – Lean-Agile Project Management
- **Agile/Scrum**
  - ☆ Agile Estimation and Analysis for Developers and Product Owners
  - ☆ Implementing Scrum for Your Team
  - ☆ Lean-Agile Testing Practices
  - ☆ Scrum Master Certification
  - – Agile Development Best Practices
  - – Effective Agile Programming
  - – Agile Use Case Analysis
  - – Agile Project Management
  - – Agile Life-Cycle Management with VersionOne
  - – Agile Software Development with Design Patterns
  - – Agile Software Development Simulation
  - – Scrum Simulation
- **Design Patterns**
  - ☆ Design Patterns Explained
  - ☆ Advanced Software Design
  - – Agile Software Development with Design Patterns
  - – Design Patterns Lab in Java, C#, or C++
- **QA/Test**
  - – Lean-Agile Testing Practices
- **Test-Driven Development**
  - ☆ Test-Driven Development
  - – Test-Driven ASP.NET
- **Effective Programming**
  - – Effective Object-Oriented Analysis and Design
  - – Effective Agile Programming
- **Object-Oriented Language-Specific Training**
  - ☆ ASP.NET
  - – C# for Java and C++ Developers
  - – C# Project-Based Training
  - – Design Patterns Lab in Java, C#, or C++
  - – Effective Object-Oriented Analysis and Design in VB.NET, C#, C++, or Java
  - – Effective ASP.NET
  - – Effective .NET with C# and ASP.NET
  - – Introduction to C++ Language and Object-Oriented Programming for C Programmers
  - – Introduction to Java Language and Object-Oriented Programming
  - – Test-Driven ASP.NET

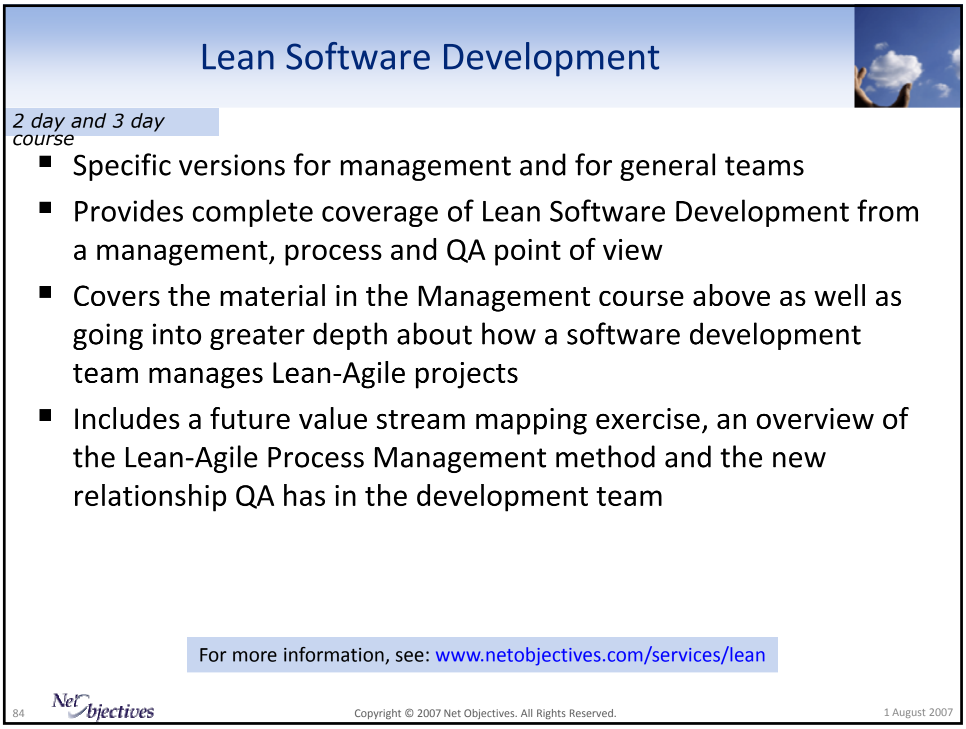
☆A Top 10 Course

For more information, see: www.netobjectives.com/services

info@netobjectives.com
www.netobjectives.com

Net Objectives Course Catalogue

Following is a list of all of the courses we offer



Lean Software Development

2 day and 3 day course

- Specific versions for management and for general teams
- Provides complete coverage of Lean Software Development from a management, process and QA point of view
- Covers the material in the Management course above as well as going into greater depth about how a software development team manages Lean-Agile projects
- Includes a future value stream mapping exercise, an overview of the Lean-Agile Process Management method and the new relationship QA has in the development team

For more information, see: www.netobjectives.com/services/lean

Lean-Agile Testing Practices

2 day course

- Lean-Agile methods promote the rapid delivery of value to the customer. One way they do this is to defer detailed definition and design of system features until the "last responsible moment." This challenges the whole team to stay continuously synchronized within very short Iteration cycles. The team must be creative, smart, and efficient with their verification and validation testing activities.
- How Lean principles can add value to your organization and how they apply to Quality Assurance goals and activities.
- Learn Agile Testing Practices needed to quickly deliver the highest value to your customers
- Discover solutions to your Lean-Agile testing challenges
- Recognize Lean opportunities for change in your organization
- Respond and adapt to Agile development changes effecting your team
- Evaluate and implement Lean-Agile testing practices for your team across the whole release cycle
- Evaluate FitNesse as an automated acceptance testing tool solution for your team
- Recommend Agile Testing transitioning solutions

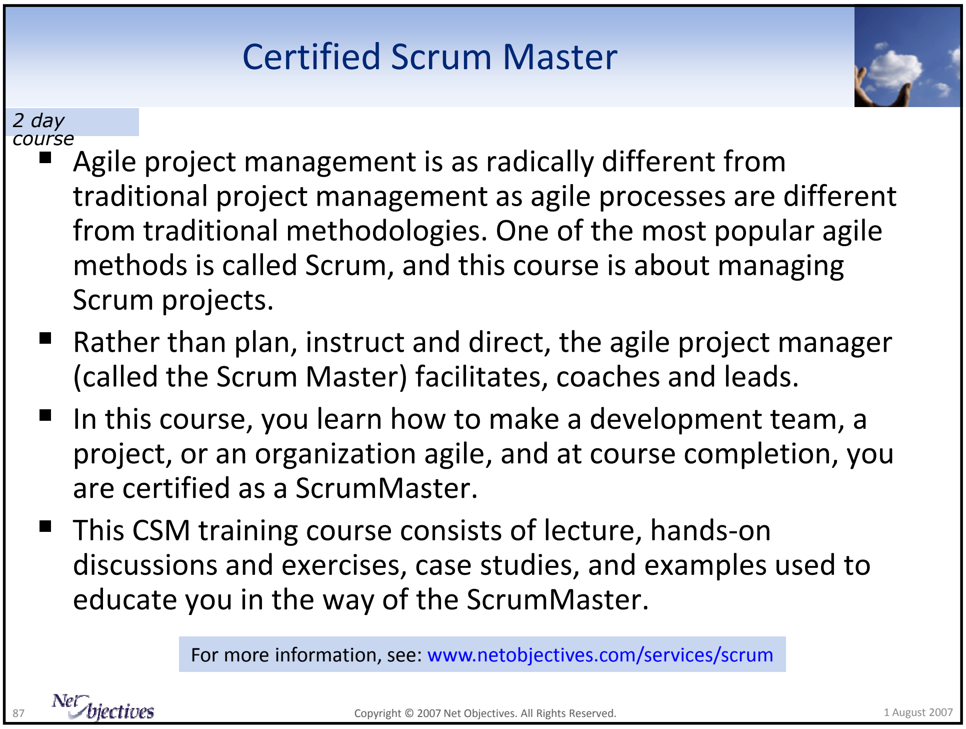
For more information, see: www.netobjectives.com/services/lean

Implementing Scrum for Your Team

2 day course

- This course is a team-centered offering that teaches a development team how to implement Scrum. It is a combination of interactive lecture with a significant amount of time spent on hands-on exercises.
- If at all possible, the entire team should attend the course together.
  - While only one or two team-members need to know how to play the role of the ScrumMaster, all the members need to know what Scrum is and what is expected of them.
- This course teaches:
  - What Scrum is
  - How to manage Scrum projects
  - How to manage requirements in Scrum projects
  - How to use Planning Poker to do story estimation
  - The roles of a Scrum team
  - The role of the ScrumMaster
  - The role of the Product Owner
  - The limitations of Scrum
  - How to Scale Scrum

For more information, see: www.netobjectives.com/services/scrum

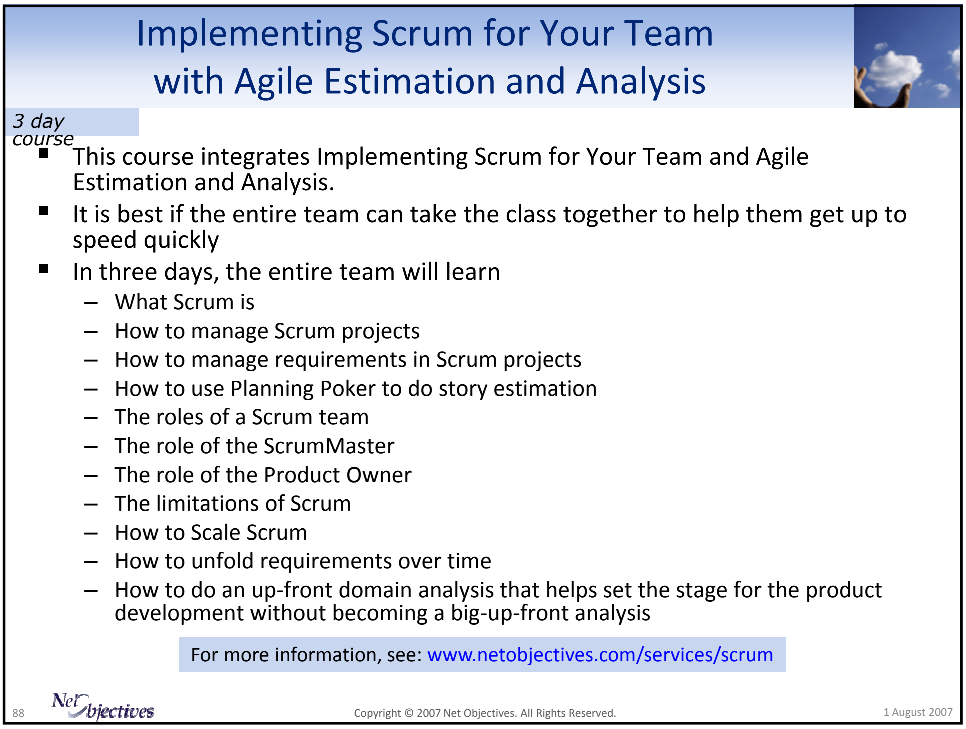


Certified Scrum Master

2 day course

- Agile project management is as radically different from traditional project management as agile processes are different from traditional methodologies. One of the most popular agile methods is called Scrum, and this course is about managing Scrum projects.
- Rather than plan, instruct and direct, the agile project manager (called the Scrum Master) facilitates, coaches and leads.
- In this course, you learn how to make a development team, a project, or an organization agile, and at course completion, you are certified as a ScrumMaster.
- This CSM training course consists of lecture, hands-on discussions and exercises, case studies, and examples used to educate you in the way of the ScrumMaster.

For more information, see: www.netobjectives.com/services/scrum

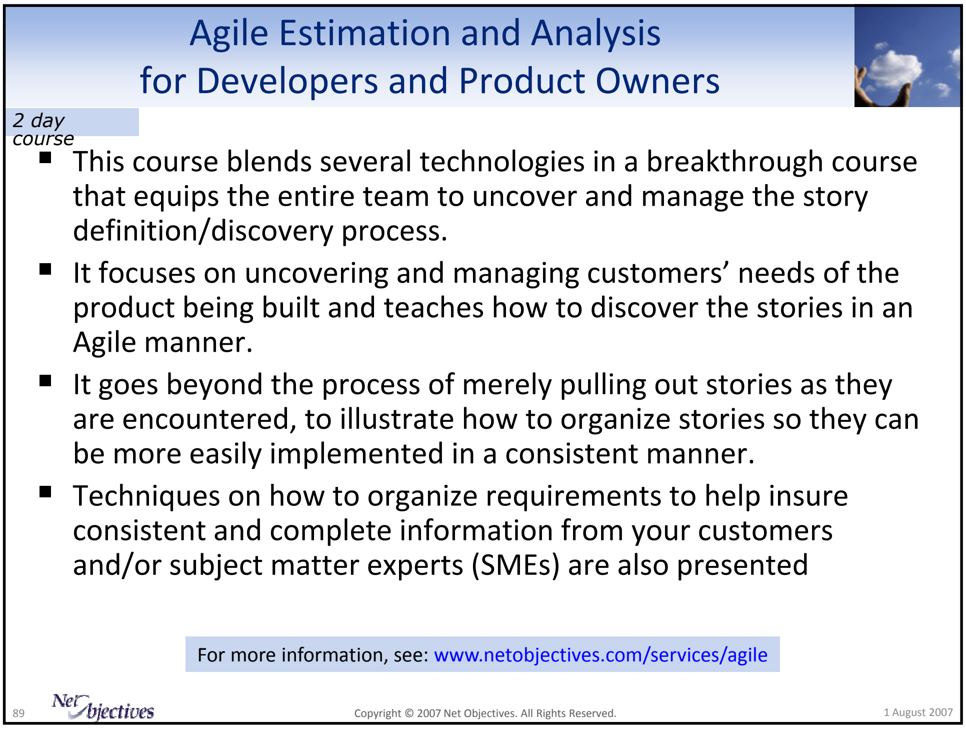


Implementing Scrum for Your Team with Agile Estimation and Analysis

3 day course

- This course integrates Implementing Scrum for Your Team and Agile Estimation and Analysis.
- It is best if the entire team can take the class together to help them get up to speed quickly
- In three days, the entire team will learn
  - What Scrum is
  - How to manage Scrum projects
  - How to manage requirements in Scrum projects
  - How to use Planning Poker to do story estimation
  - The roles of a Scrum team
  - The role of the ScrumMaster
  - The role of the Product Owner
  - The limitations of Scrum
  - How to Scale Scrum
  - How to unfold requirements over time
  - How to do an up-front domain analysis that helps set the stage for the product development without becoming a big-up-front analysis

For more information, see: www.netobjectives.com/services/scrum

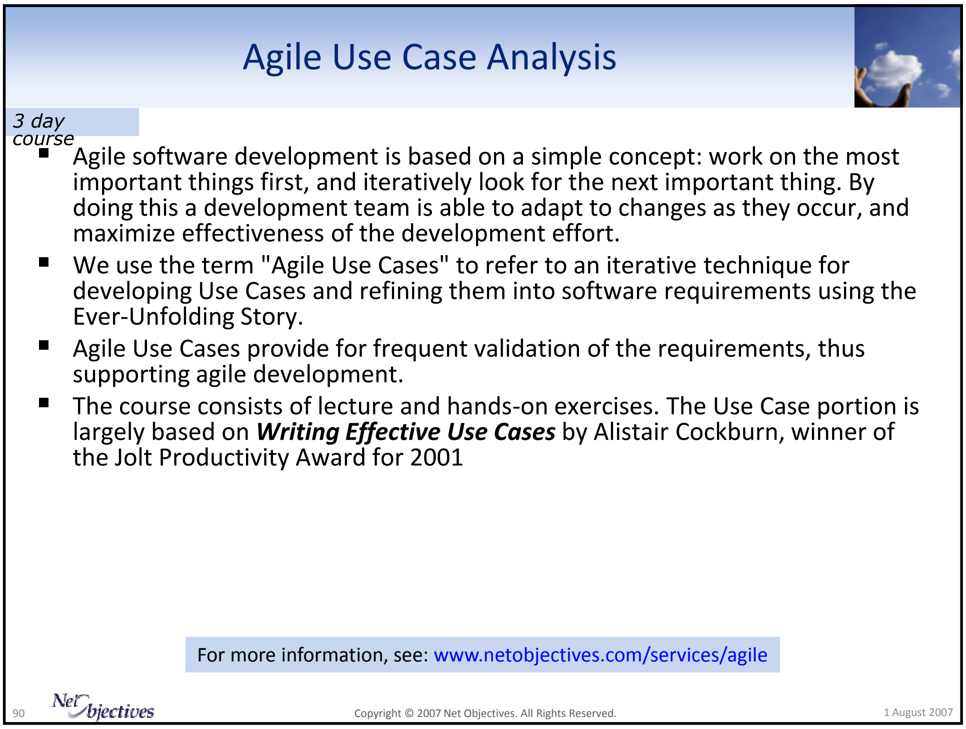


Agile Estimation and Analysis for Developers and Product Owners

2 day course

- This course blends several technologies in a breakthrough course that equips the entire team to uncover and manage the story definition/discovery process.
- It focuses on uncovering and managing customers' needs of the product being built and teaches how to discover the stories in an Agile manner.
- It goes beyond the process of merely pulling out stories as they are encountered, to illustrate how to organize stories so they can be more easily implemented in a consistent manner.
- Techniques on how to organize requirements to help insure consistent and complete information from your customers and/or subject matter experts (SMEs) are also presented

For more information, see: www.netobjectives.com/services/agile

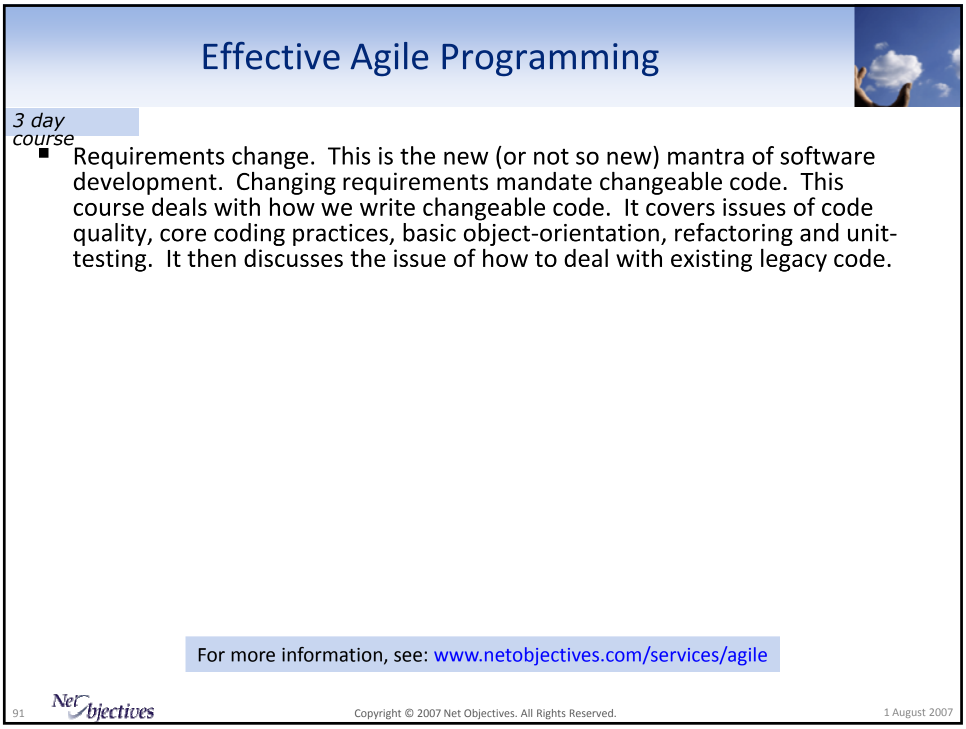


Agile Use Case Analysis

3 day course

- Agile software development is based on a simple concept: work on the most important things first, and iteratively look for the next important thing. By doing this a development team is able to adapt to changes as they occur, and maximize effectiveness of the development effort.
- We use the term "Agile Use Cases" to refer to an iterative technique for developing Use Cases and refining them into software requirements using the Ever-Unfolding Story.
- Agile Use Cases provide for frequent validation of the requirements, thus supporting agile development.
- The course consists of lecture and hands-on exercises. The Use Case portion is largely based on ***Writing Effective Use Cases*** by Alistair Cockburn, winner of the Jolt Productivity Award for 2001

For more information, see: www.netobjectives.com/services/agile

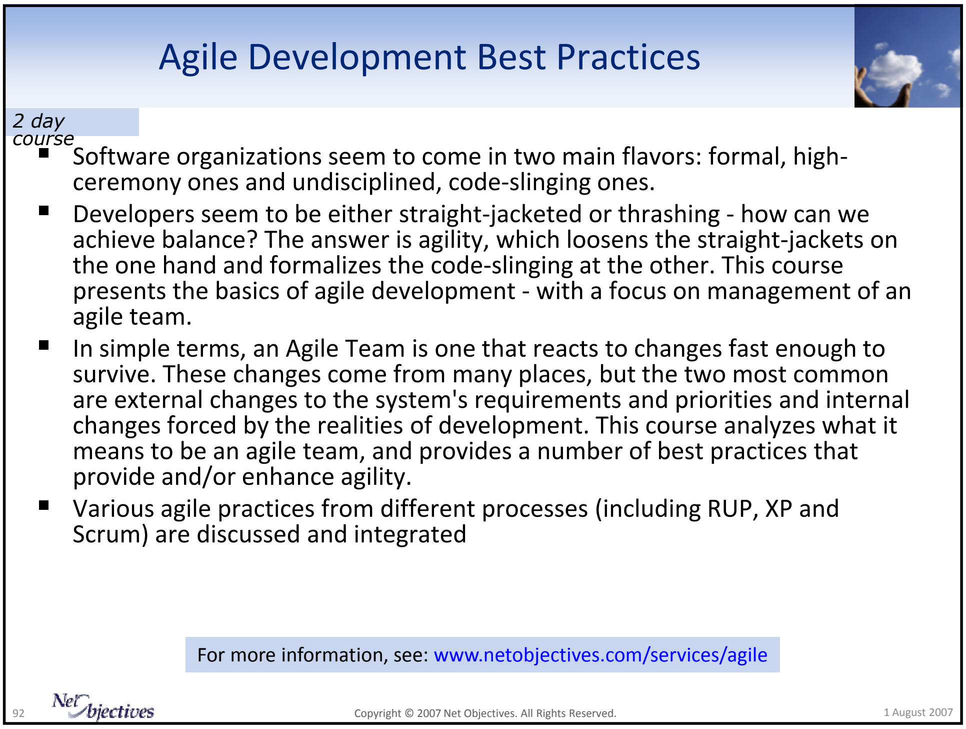


Effective Agile Programming

3 day course

- Requirements change. This is the new (or not so new) mantra of software development. Changing requirements mandate changeable code. This course deals with how we write changeable code. It covers issues of code quality, core coding practices, basic object-orientation, refactoring and unit-testing. It then discusses the issue of how to deal with existing legacy code.

For more information, see: www.netobjectives.com/services/agile



Agile Development Best Practices

2 day course

- Software organizations seem to come in two main flavors: formal, high-ceremony ones and undisciplined, code-slinging ones.
- Developers seem to be either straight-jacketed or thrashing - how can we achieve balance? The answer is agility, which loosens the straight-jackets on the one hand and formalizes the code-slinging at the other. This course presents the basics of agile development - with a focus on management of an agile team.
- In simple terms, an Agile Team is one that reacts to changes fast enough to survive. These changes come from many places, but the two most common are external changes to the system's requirements and priorities and internal changes forced by the realities of development. This course analyzes what it means to be an agile team, and provides a number of best practices that provide and/or enhance agility.
- Various agile practices from different processes (including RUP, XP and Scrum) are discussed and integrated

For more information, see: www.netobjectives.com/services/agile

Agile Life-Cycle Management with VersionOne

2 day course

- Provides the necessary understanding of how Agile projects are managed and how to use VersionOne's flagship product ***V1: Agile Enterprise*** to manage one's work.
- VersionOne's 100% web-based management platform incorporates a simple, intuitive framework for organizations introducing and scaling their agile development efforts.
- Using VersionOne, all project stakeholders - developers, testers, managers, product managers, customers, and software executives - work together to easily coordinate project plans, priorities, and progress.
- Deployable in minutes, VersionOne enables development teams to accelerate the rollout of today's leading agile methodologies across multiple projects, releases, teams, and locations. Configurable, methodology-specific templates for Scrum, Extreme Programming (XP), DSDM, and Agile UP accelerate internal deployment and can be customized for teams utilizing hybrid methodologies.

VERSIONONE
Simplifying Software Delivery

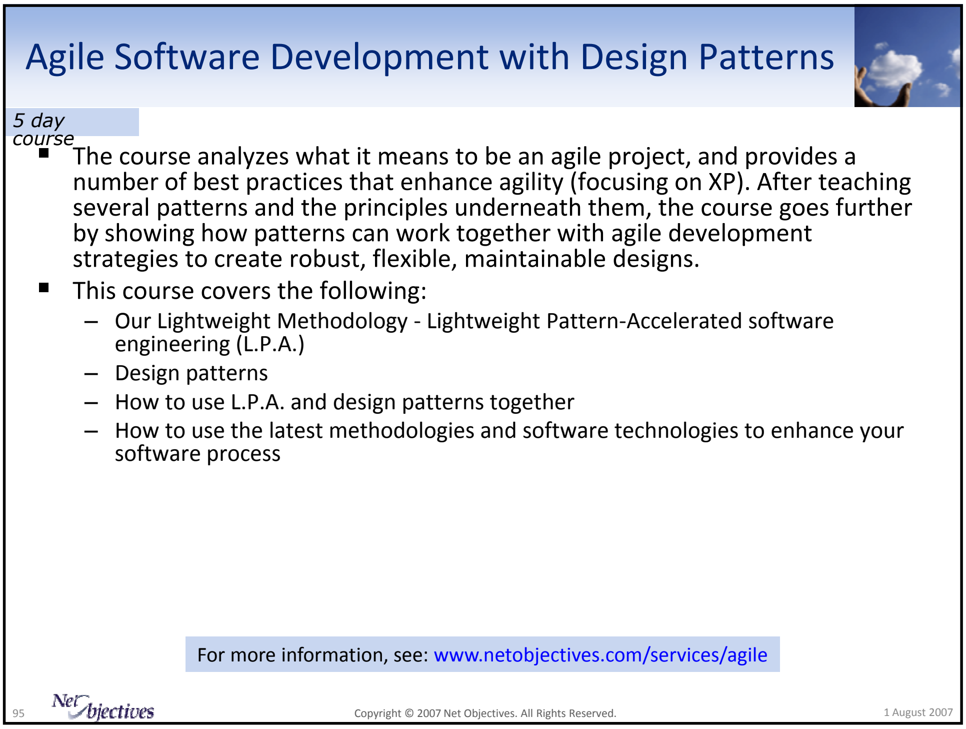
For more information, see: www.netobjectives.com/services/agile

Agile Project Management

2 day course

- Managing software organizations is difficult, as there are many different project personalities. Agile teams are teams that actively identify changes in requirements and priorities and evolve their product in response to them.
- Traditional Project Managers follow the maxim "plan your work, then work your plan" while an agile Project Manager has the mantra "produce maximum value for money spent". The processes and practices discussed in this course are primarily Scrum, but have inputs from a variety of other sources, most notably RUP and XP
- This course is intended for teams that intend on becoming more agile. The course's purpose is to introduce the basic concepts to team members so that their transition to agility is based on some consistency. This course is not a course specifically for agile project managers; for that you should take either (or both) of the Certified ScrumMaster and Agile Estimation and Analysis for Developers and Product Owners courses.
- The course consists of lecture, discussion, and many hands-on exercises, and presents the basics of agile Project Management as practiced in Scrum - a small-team agile method.

For more information, see: www.netobjectives.com/services/agile

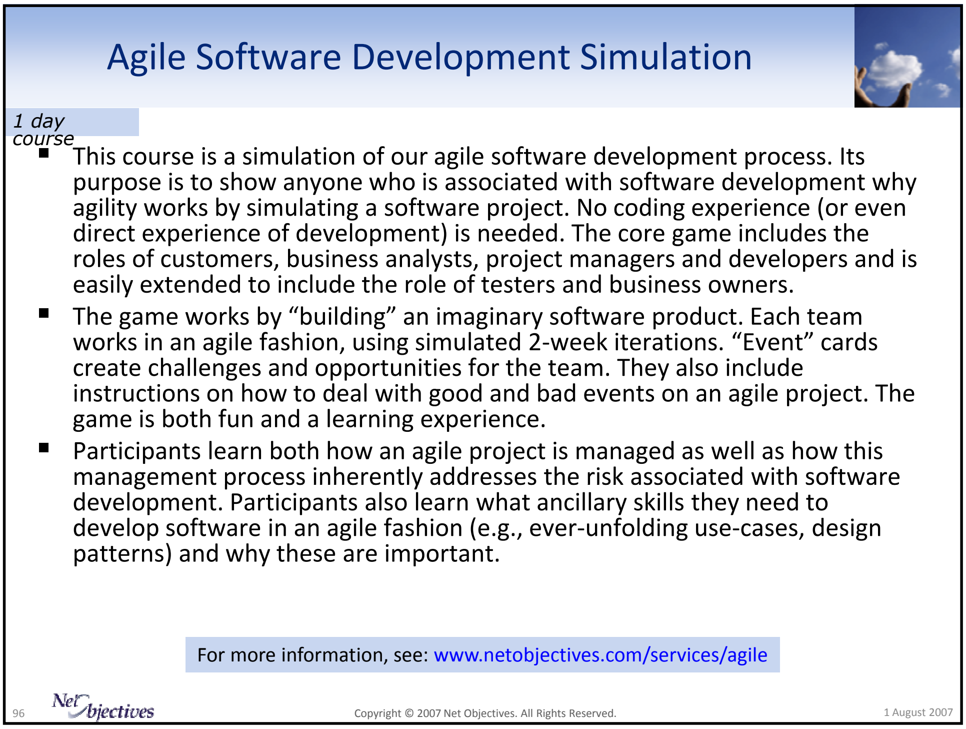


Agile Software Development with Design Patterns

5 day course

- The course analyzes what it means to be an agile project, and provides a number of best practices that enhance agility (focusing on XP). After teaching several patterns and the principles underneath them, the course goes further by showing how patterns can work together with agile development strategies to create robust, flexible, maintainable designs.
- This course covers the following:
  - Our Lightweight Methodology - Lightweight Pattern-Accelerated software engineering (L.P.A.)
  - Design patterns
  - How to use L.P.A. and design patterns together
  - How to use the latest methodologies and software technologies to enhance your software process

For more information, see: www.netobjectives.com/services/agile

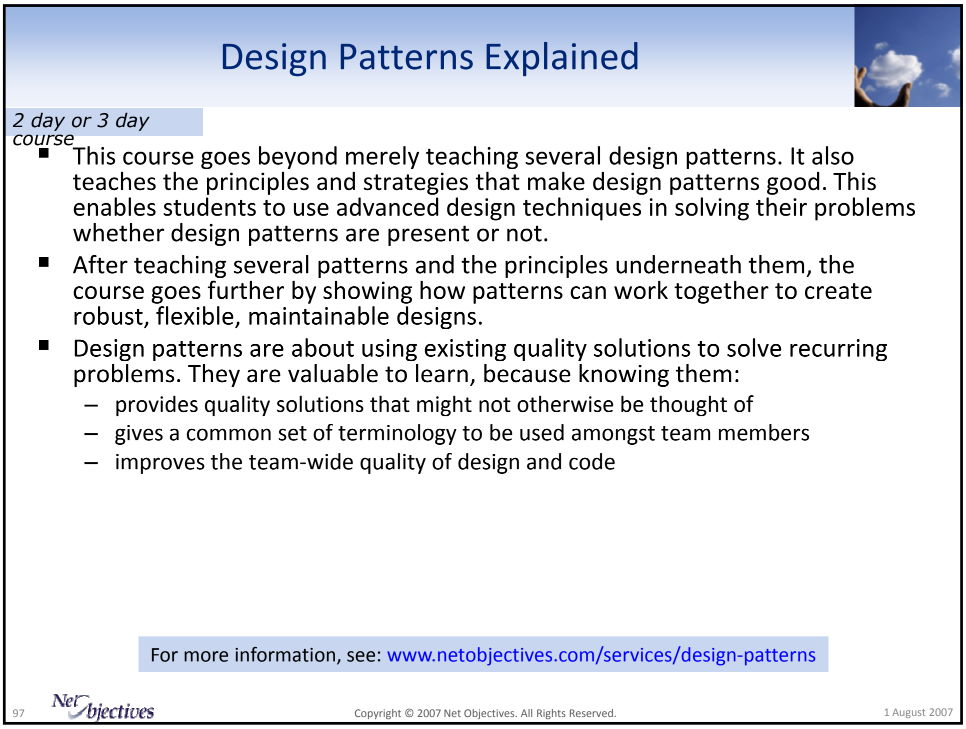


Agile Software Development Simulation

1 day course

- This course is a simulation of our agile software development process. Its purpose is to show anyone who is associated with software development why agility works by simulating a software project. No coding experience (or even direct experience of development) is needed. The core game includes the roles of customers, business analysts, project managers and developers and is easily extended to include the role of testers and business owners.
- The game works by “building” an imaginary software product. Each team works in an agile fashion, using simulated 2-week iterations. “Event” cards create challenges and opportunities for the team. They also include instructions on how to deal with good and bad events on an agile project. The game is both fun and a learning experience.
- Participants learn both how an agile project is managed as well as how this management process inherently addresses the risk associated with software development. Participants also learn what ancillary skills they need to develop software in an agile fashion (e.g., ever-unfolding use-cases, design patterns) and why these are important.

For more information, see: www.netobjectives.com/services/agile

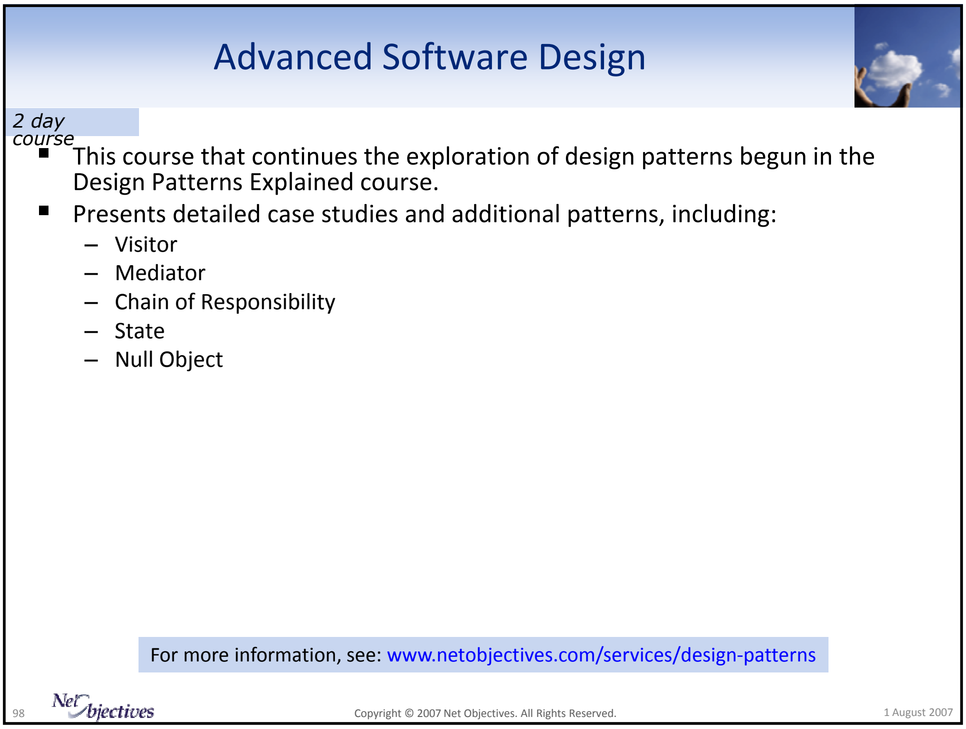


Design Patterns Explained

2 day or 3 day course

- This course goes beyond merely teaching several design patterns. It also teaches the principles and strategies that make design patterns good. This enables students to use advanced design techniques in solving their problems whether design patterns are present or not.
- After teaching several patterns and the principles underneath them, the course goes further by showing how patterns can work together to create robust, flexible, maintainable designs.
- Design patterns are about using existing quality solutions to solve recurring problems. They are valuable to learn, because knowing them:
  - provides quality solutions that might not otherwise be thought of
  - gives a common set of terminology to be used amongst team members
  - improves the team-wide quality of design and code

For more information, see: www.netobjectives.com/services/design-patterns

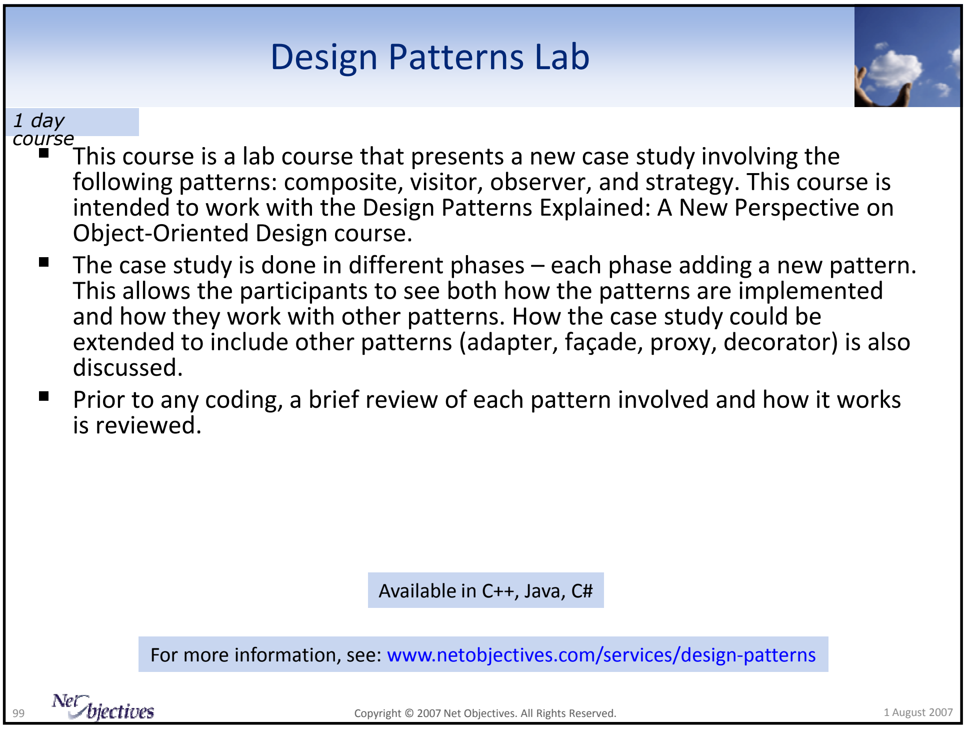


Advanced Software Design

2 day course

- This course that continues the exploration of design patterns begun in the Design Patterns Explained course.
- Presents detailed case studies and additional patterns, including:
  - Visitor
  - Mediator
  - Chain of Responsibility
  - State
  - Null Object

For more information, see: www.netobjectives.com/services/design-patterns



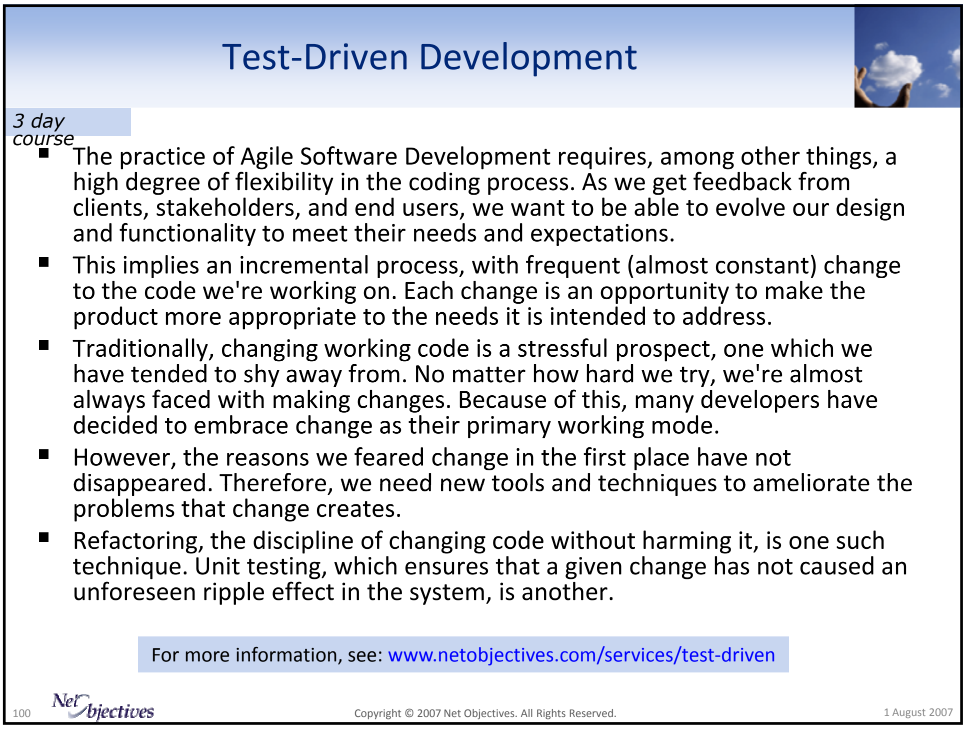
Design Patterns Lab

1 day course

- This course is a lab course that presents a new case study involving the following patterns: composite, visitor, observer, and strategy. This course is intended to work with the Design Patterns Explained: A New Perspective on Object-Oriented Design course.
- The case study is done in different phases – each phase adding a new pattern. This allows the participants to see both how the patterns are implemented and how they work with other patterns. How the case study could be extended to include other patterns (adapter, façade, proxy, decorator) is also discussed.
- Prior to any coding, a brief review of each pattern involved and how it works is reviewed.

Available in C++, Java, C#

For more information, see: www.netobjectives.com/services/design-patterns

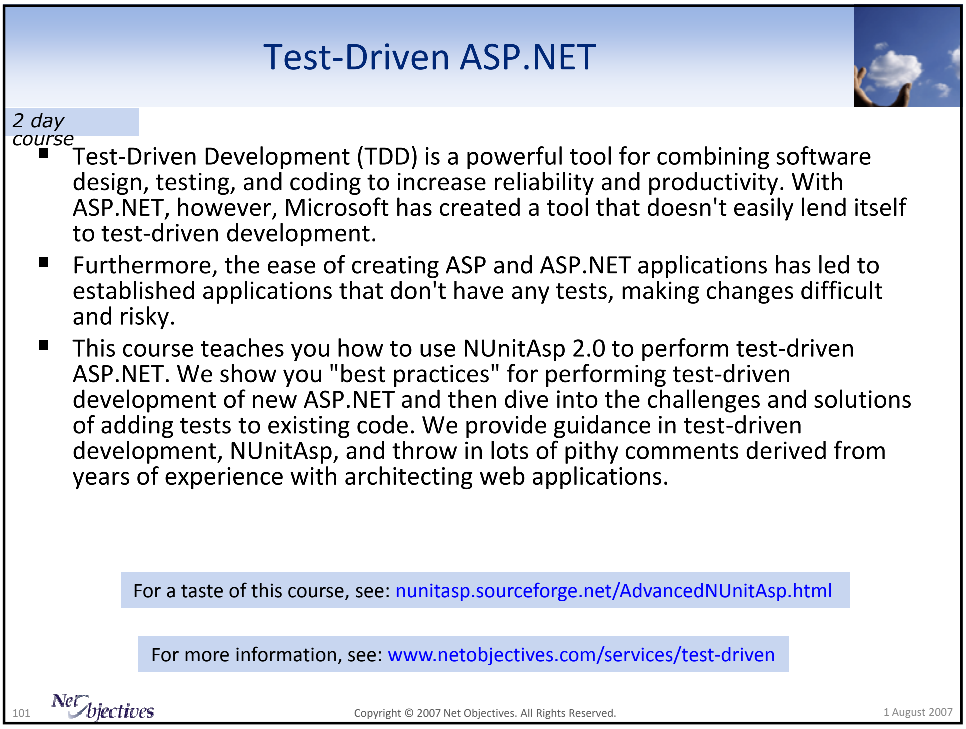


Test-Driven Development

3 day course

- The practice of Agile Software Development requires, among other things, a high degree of flexibility in the coding process. As we get feedback from clients, stakeholders, and end users, we want to be able to evolve our design and functionality to meet their needs and expectations.
- This implies an incremental process, with frequent (almost constant) change to the code we're working on. Each change is an opportunity to make the product more appropriate to the needs it is intended to address.
- Traditionally, changing working code is a stressful prospect, one which we have tended to shy away from. No matter how hard we try, we're almost always faced with making changes. Because of this, many developers have decided to embrace change as their primary working mode.
- However, the reasons we feared change in the first place have not disappeared. Therefore, we need new tools and techniques to ameliorate the problems that change creates.
- Refactoring, the discipline of changing code without harming it, is one such technique. Unit testing, which ensures that a given change has not caused an unforeseen ripple effect in the system, is another.

For more information, see: www.netobjectives.com/services/test-driven



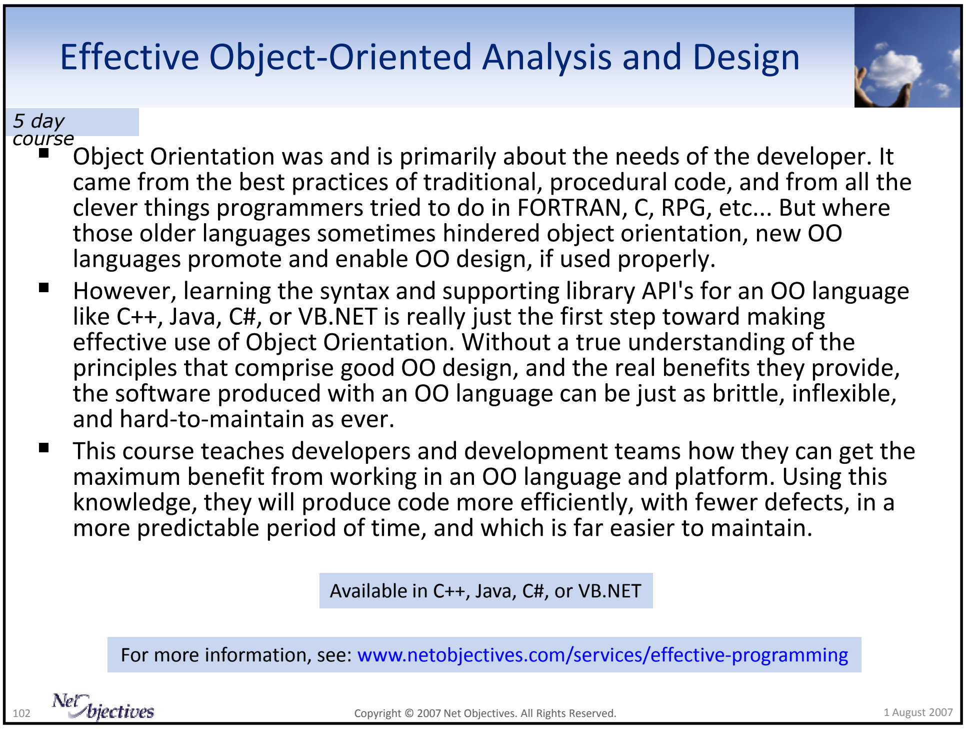
Test-Driven ASP.NET

2 day course

- Test-Driven Development (TDD) is a powerful tool for combining software design, testing, and coding to increase reliability and productivity. With ASP.NET, however, Microsoft has created a tool that doesn't easily lend itself to test-driven development.
- Furthermore, the ease of creating ASP and ASP.NET applications has led to established applications that don't have any tests, making changes difficult and risky.
- This course teaches you how to use NUnitAsp 2.0 to perform test-driven ASP.NET. We show you "best practices" for performing test-driven development of new ASP.NET and then dive into the challenges and solutions of adding tests to existing code. We provide guidance in test-driven development, NUnitAsp, and throw in lots of pithy comments derived from years of experience with architecting web applications.

For a taste of this course, see: nunitasp.sourceforge.net/AdvancedNUnitAsp.html

For more information, see: www.netobjectives.com/services/test-driven



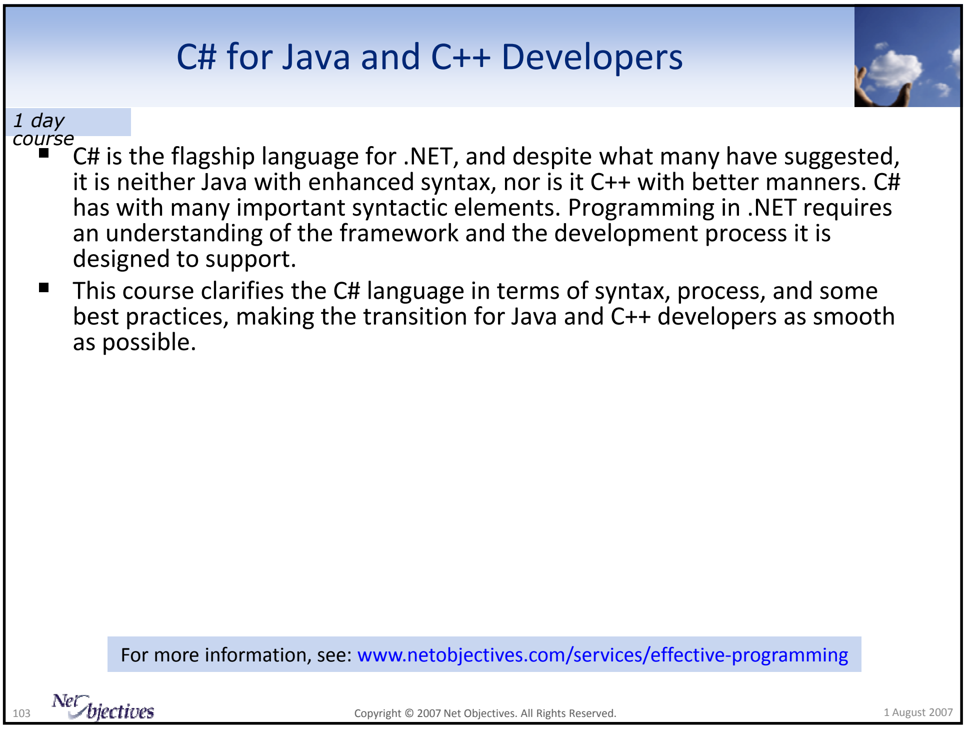
Effective Object-Oriented Analysis and Design

5 day course

- Object Orientation was and is primarily about the needs of the developer. It came from the best practices of traditional, procedural code, and from all the clever things programmers tried to do in FORTRAN, C, RPG, etc... But where those older languages sometimes hindered object orientation, new OO languages promote and enable OO design, if used properly.
- However, learning the syntax and supporting library API's for an OO language like C++, Java, C#, or VB.NET is really just the first step toward making effective use of Object Orientation. Without a true understanding of the principles that comprise good OO design, and the real benefits they provide, the software produced with an OO language can be just as brittle, inflexible, and hard-to-maintain as ever.
- This course teaches developers and development teams how they can get the maximum benefit from working in an OO language and platform. Using this knowledge, they will produce code more efficiently, with fewer defects, in a more predictable period of time, and which is far easier to maintain.

Available in C++, Java, C#, or VB.NET

For more information, see: www.netobjectives.com/services/effective-programming

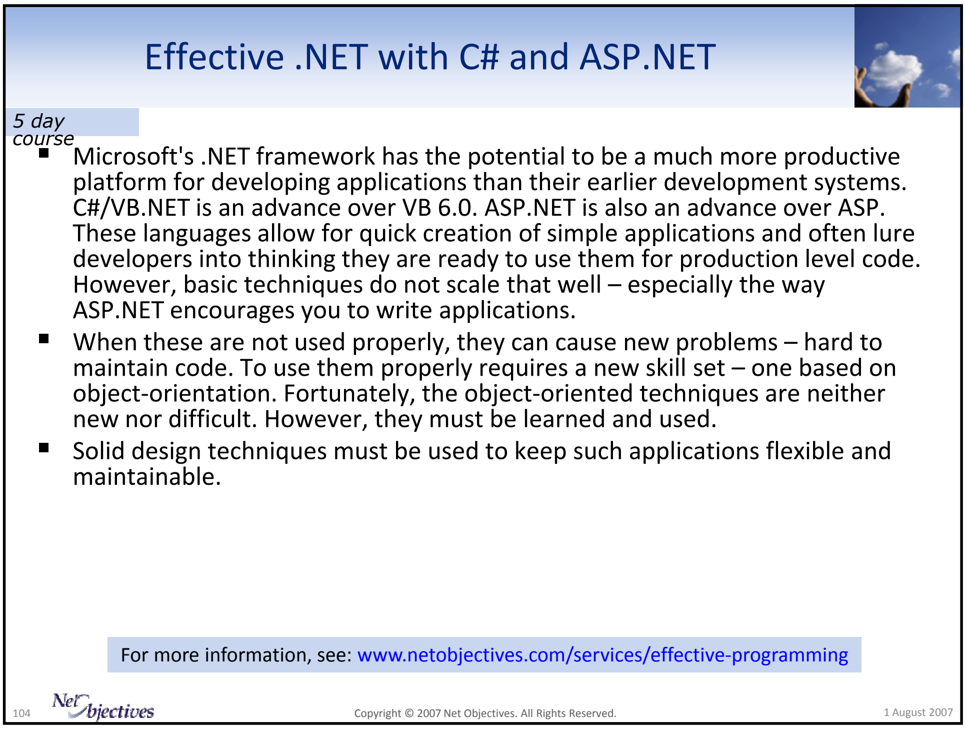


C# for Java and C++ Developers

1 day course

- C# is the flagship language for .NET, and despite what many have suggested, it is neither Java with enhanced syntax, nor is it C++ with better manners. C# has with many important syntactic elements. Programming in .NET requires an understanding of the framework and the development process it is designed to support.
- This course clarifies the C# language in terms of syntax, process, and some best practices, making the transition for Java and C++ developers as smooth as possible.

For more information, see: www.netobjectives.com/services/effective-programming

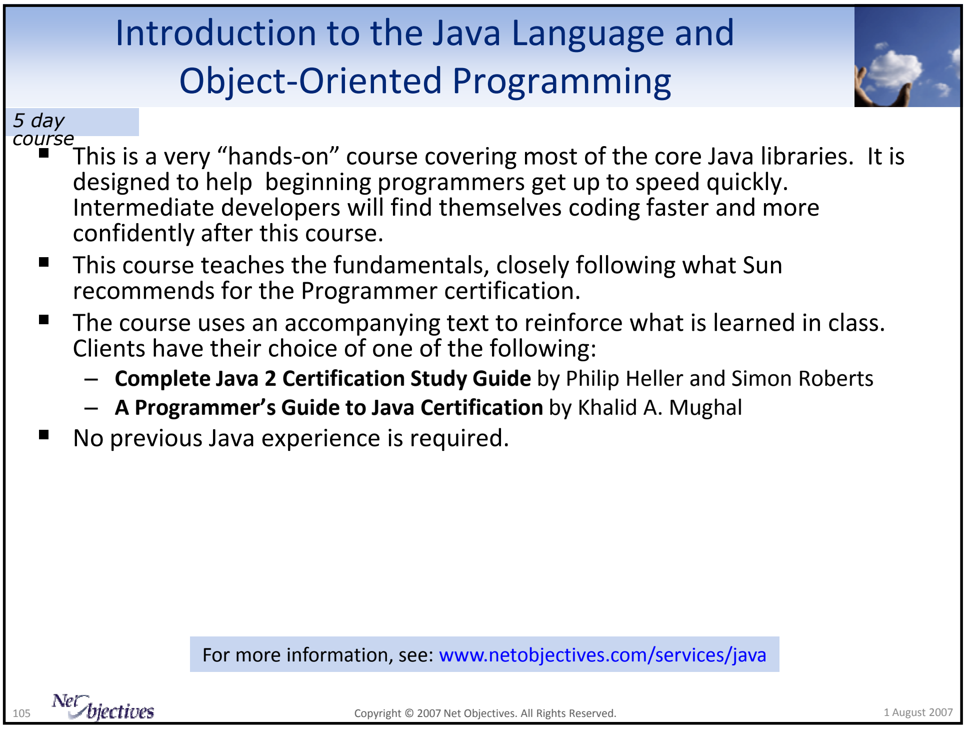


Effective .NET with C# and ASP.NET

5 day course

- Microsoft's .NET framework has the potential to be a much more productive platform for developing applications than their earlier development systems. C#/VB.NET is an advance over VB 6.0. ASP.NET is also an advance over ASP. These languages allow for quick creation of simple applications and often lure developers into thinking they are ready to use them for production level code. However, basic techniques do not scale that well – especially the way ASP.NET encourages you to write applications.
- When these are not used properly, they can cause new problems – hard to maintain code. To use them properly requires a new skill set – one based on object-orientation. Fortunately, the object-oriented techniques are neither new nor difficult. However, they must be learned and used.
- Solid design techniques must be used to keep such applications flexible and maintainable.

For more information, see: www.netobjectives.com/services/effective-programming

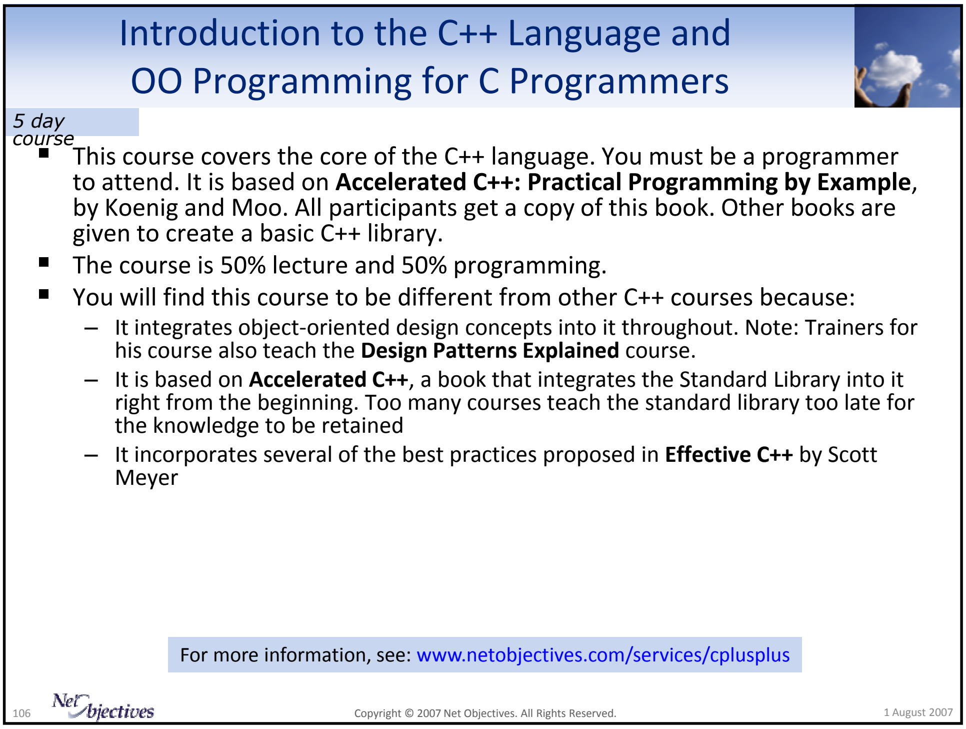


Introduction to the Java Language and Object-Oriented Programming

5 day course

- This is a very “hands-on” course covering most of the core Java libraries. It is designed to help beginning programmers get up to speed quickly. Intermediate developers will find themselves coding faster and more confidently after this course.
- This course teaches the fundamentals, closely following what Sun recommends for the Programmer certification.
- The course uses an accompanying text to reinforce what is learned in class. Clients have their choice of one of the following:
  - **Complete Java 2 Certification Study Guide** by Philip Heller and Simon Roberts
  - **A Programmer’s Guide to Java Certification** by Khalid A. Mughal
- No previous Java experience is required.

For more information, see: www.netobjectives.com/services/java

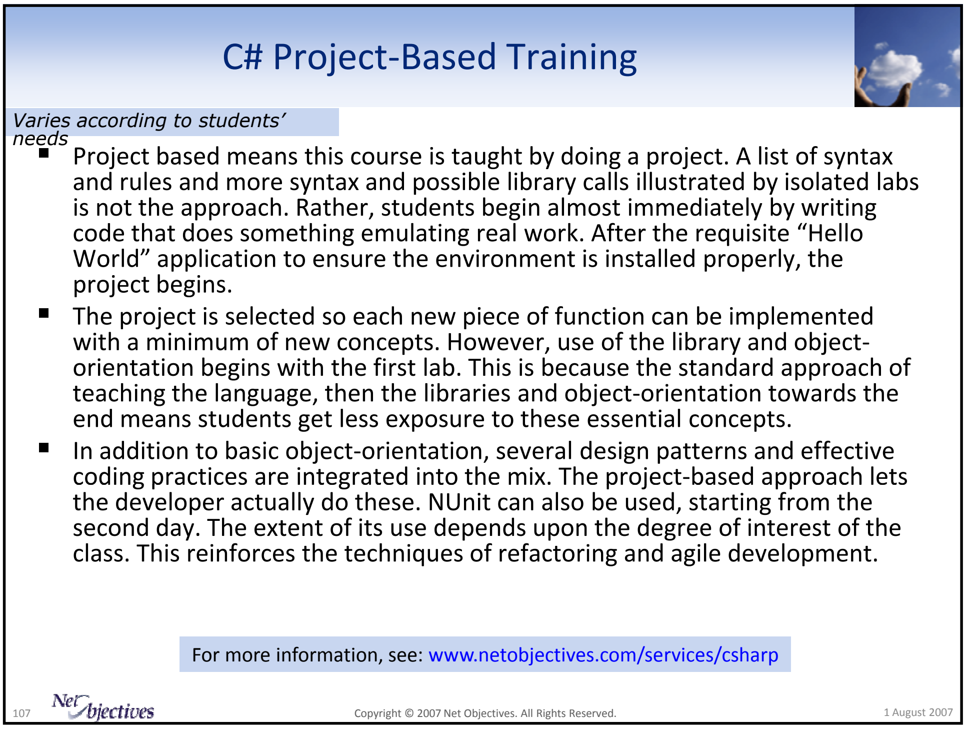


Introduction to the C++ Language and OO Programming for C Programmers

5 day course

- This course covers the core of the C++ language. You must be a programmer to attend. It is based on **Accelerated C++: Practical Programming by Example**, by Koenig and Moo. All participants get a copy of this book. Other books are given to create a basic C++ library.
- The course is 50% lecture and 50% programming.
- You will find this course to be different from other C++ courses because:
  - It integrates object-oriented design concepts into it throughout. Note: Trainers for his course also teach the **Design Patterns Explained** course.
  - It is based on **Accelerated C++**, a book that integrates the Standard Library into it right from the beginning. Too many courses teach the standard library too late for the knowledge to be retained
  - It incorporates several of the best practices proposed in **Effective C++** by Scott Meyer

For more information, see: www.netobjectives.com/services/cplusplus

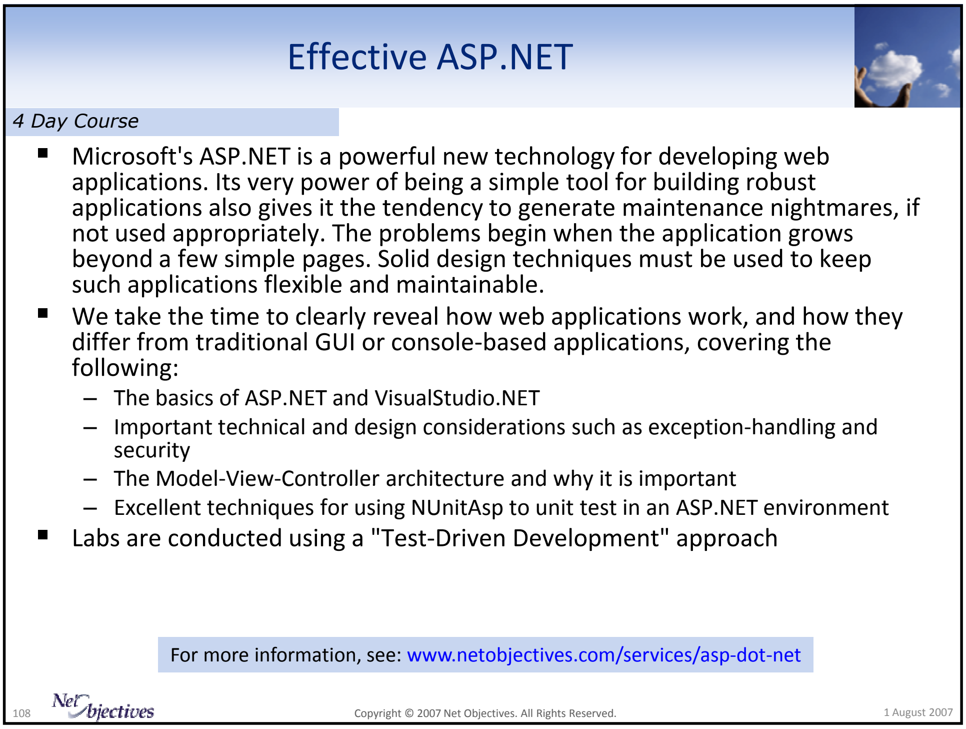


C# Project-Based Training

Varies according to students' needs

- Project based means this course is taught by doing a project. A list of syntax and rules and more syntax and possible library calls illustrated by isolated labs is not the approach. Rather, students begin almost immediately by writing code that does something emulating real work. After the requisite "Hello World" application to ensure the environment is installed properly, the project begins.
- The project is selected so each new piece of function can be implemented with a minimum of new concepts. However, use of the library and object-orientation begins with the first lab. This is because the standard approach of teaching the language, then the libraries and object-orientation towards the end means students get less exposure to these essential concepts.
- In addition to basic object-orientation, several design patterns and effective coding practices are integrated into the mix. The project-based approach lets the developer actually do these. NUnit can also be used, starting from the second day. The extent of its use depends upon the degree of interest of the class. This reinforces the techniques of refactoring and agile development.

For more information, see: www.netobjectives.com/services/csharp



Effective ASP.NET

4 Day Course

- Microsoft's ASP.NET is a powerful new technology for developing web applications. Its very power of being a simple tool for building robust applications also gives it the tendency to generate maintenance nightmares, if not used appropriately. The problems begin when the application grows beyond a few simple pages. Solid design techniques must be used to keep such applications flexible and maintainable.
- We take the time to clearly reveal how web applications work, and how they differ from traditional GUI or console-based applications, covering the following:
  - The basics of ASP.NET and VisualStudio.NET
  - Important technical and design considerations such as exception-handling and security
  - The Model-View-Controller architecture and why it is important
  - Excellent techniques for using NUnitAsp to unit test in an ASP.NET environment
- Labs are conducted using a "Test-Driven Development" approach

For more information, see: www.netobjectives.com/services/asp-dot-net