Agile with XP and Scrum

Amit Goel

National Agile Software Workshop @ Indore

Agile India Conference

Agile Software Community of India

Disclaimer and Credits

Most of material in this presentation has been
inspired (please read as "reused") from a number of sources,

I take it as
"Don't reinvent the wheel"
and
"Spreading the good words around"

The roll of honor for Credits:

Scott Ambler   XP

Martin Fowler   Scrum

Alistair Cockburn   Thoughtworks

Craig Larman   Kent Beck

Jim Highsmith

Rules of the Game

- Mobile Phones
  - Keep your mobile phones in silent mode
  - Sit near exit if you expect to interrupt the session for whatsoever
- Side-talks
  - Side conversations are a strict no-no
  - Raise your hand if you wish to make a point
- Interaction
  - Do not hesitate to ask any questions
  - Do not wait until the end of session for questions
- **Innovation**
  - **A lot of new ideas would be presented, use at your own risk**

- I am very blunt at times, kindly bear with me

Agenda

- Setting Expectations

- What is Agile

- eXtreme Programming

- Scrum

What is Agile

Software Development

- Paradigm Problem – Predictable Manufacturing
    - Waterfall cycle
    - Big up-front specs
    - Speculative plans
    - Estimations

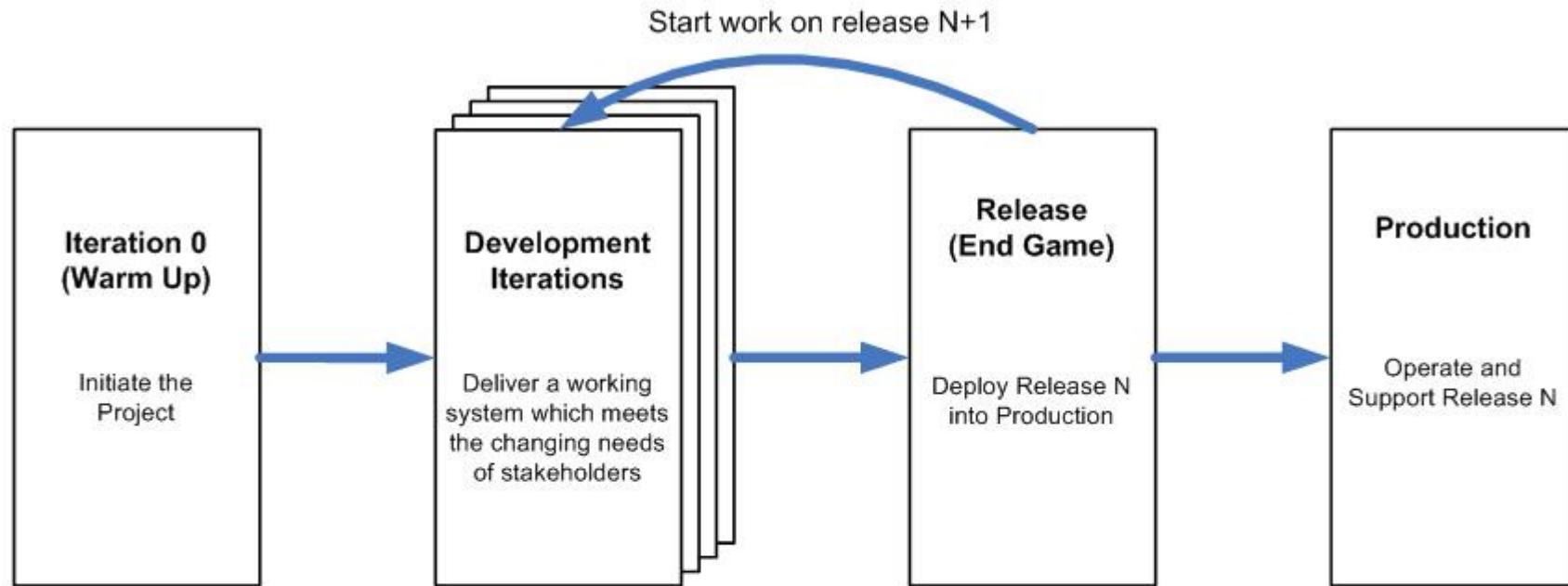- **Software Development is a Domain of Inventive, High-Change, High-Novelty Work – A Craft**

Craig Larman

Agile is an iterative and incremental (evolutionary)
approach to software development
which is performed in a highly collaborative manner
by self-organizing teams
with "just enough" ceremony
that produces high quality software
in a cost effective and timely manner
which meets the changing needs of its stakeholders.

Scott Ambler

Agile SDLC

Start work on release N+1

Iteration 0 (Warm Up)

Initiate the Project

Development Iterations

Deliver a working system which meets the changing needs of stakeholders

Release (End Game)

Deploy Release N into Production

Production

Operate and Support Release N

- Active stakeholder participation
- Obtain funding and support
- Start building the team
- Initial requirements envisioning
- Initial architecture envisioning
- Setup environment

- Active stakeholder participation
- Collaborative development
- Model storming
- Test driven design (TDD)
- Confirmatory testing
- Investigative testing
- Evolve documentation
- Internally deploy software

- Active stakeholder participation
- Final system testing
- Final acceptance testing
- Finalize documentation
- Pilot test the release
- Train end users
- Train production staff
- Deploy system into production

- Operate system
- Support system
- Identify defects and enhancements

Copyright 2006-2007 Scott W. Ambler

Agile Advantages

- Helps handle changing requirements & priorities

- Lowers cost of change

- Provides better visibility into project progress

- Reduces risk

- Maximizes return on investment (business value prioritized)

- Encourages higher quality code

- Delivers business value early & often

e**X**treme **P**rogramming

XP Map

Extreme Programming Project
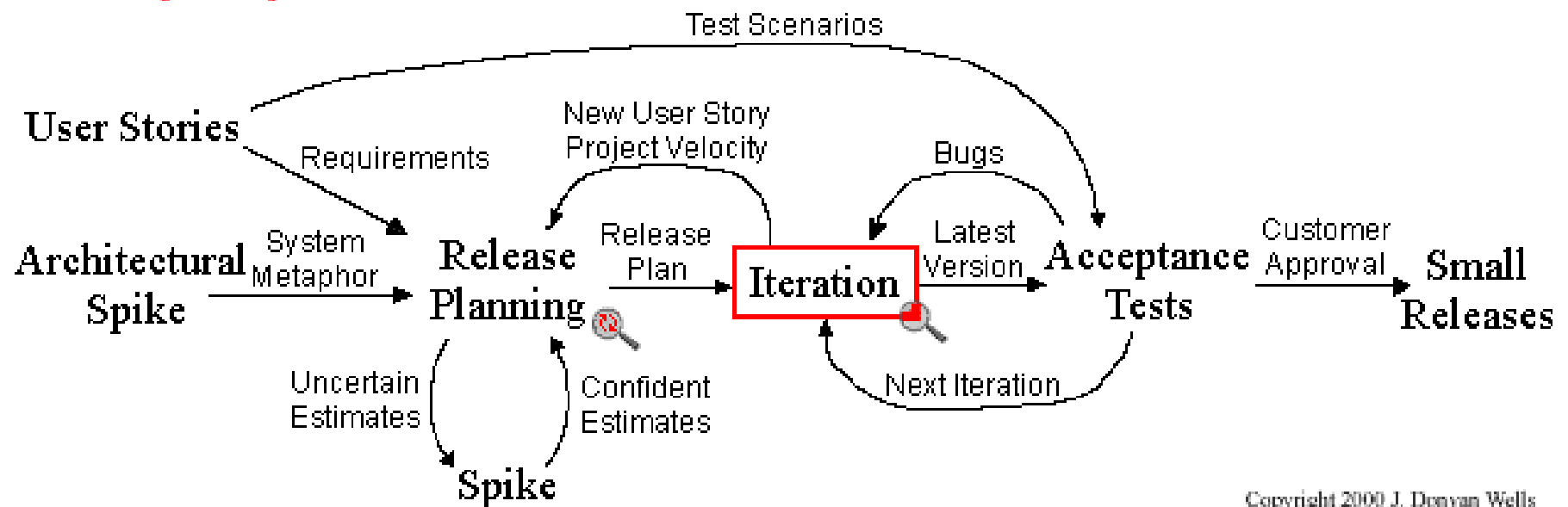
XP Values

- Communication

- Feedback

- Simplicity

- Courage

- Respect

Communication leads to valuable feedback which encourages simplicity which allows for courage to change

XP Lifecycle

- **Exploration**
    - customers write story cards, project team becomes familiar with tools, technology and practices

- **Planning**
    - set priority of stories and contents of first release

- **Iterations to Release**
    - Testing and programming in iterations, iterations planning

- **Productionizing**
    - Operational deployment, extra testing and checking before release to customer, documentation, training
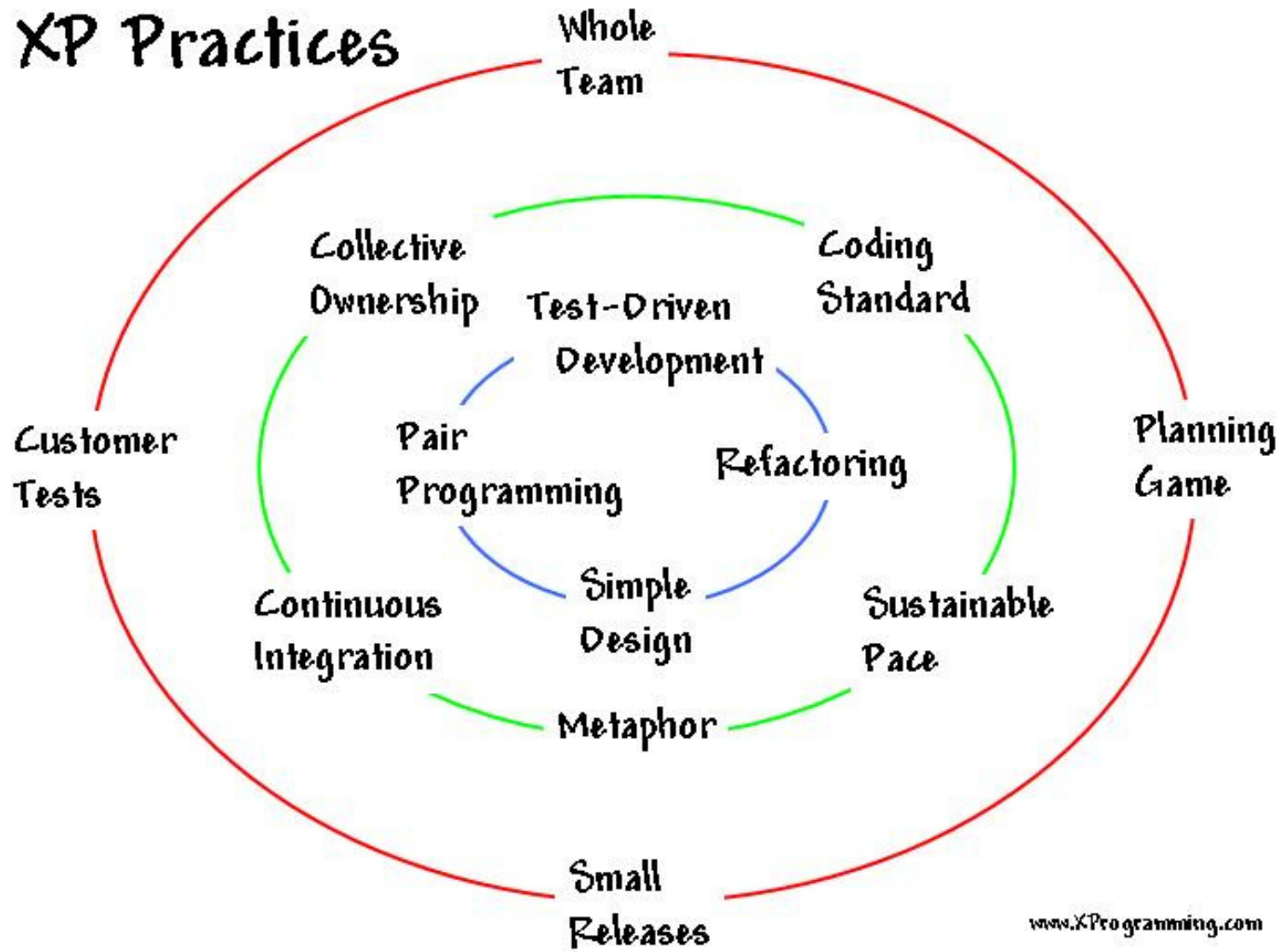
- **Maintenance**
    - Enhance, fix, could repeat phases for incremental releases

XP Roles

- **Customer**
  - Writes stories and functional tests
  - Picks stories for release and iteration
- **Programmer**
  - Writes tests and code
  - Designs and Refactors
- **Tester**
  - Helps customer write tests and runs them
- **Coach**
  - Customizes XP process
  - Makes sure everyone follows the XP process
- **Tracker**
  - Collects metrics
  - Gives feedback on estimates and process on iterations
- **Consultant**
  - supplies specific technical knowledge needed

XP Practices

Whole Team

Collective Ownership

Coding Standard

Test-Driven Development

Customer Tests

Pair Programming

Refactoring

Planning Game

Continuous Integration

Simple Design

Sustainable Pace

Metaphor

Small Releases

www.XProgramming.com

Planning Game

- **User Stories**
  - The customer cares about
  - Can be reasonably tested
  - Can be estimated and prioritized

- **The Planning Game**
  - Users write stories
  - Developers estimate them
  - Users split, merge and prioritize
  - Plan releases and the next iteration

Test Driven Development

- Customer Tests
  - Specified by user
  - Implemented by user, developer and/or test team
  - Automated
  - Part of specifications
  - Run at every release
- Unit Tests
  - Written by developers
  - Run before, during and after coding

Design

- Simple Design

- Metaphor

- Spike Solutions

- Functionality not added early

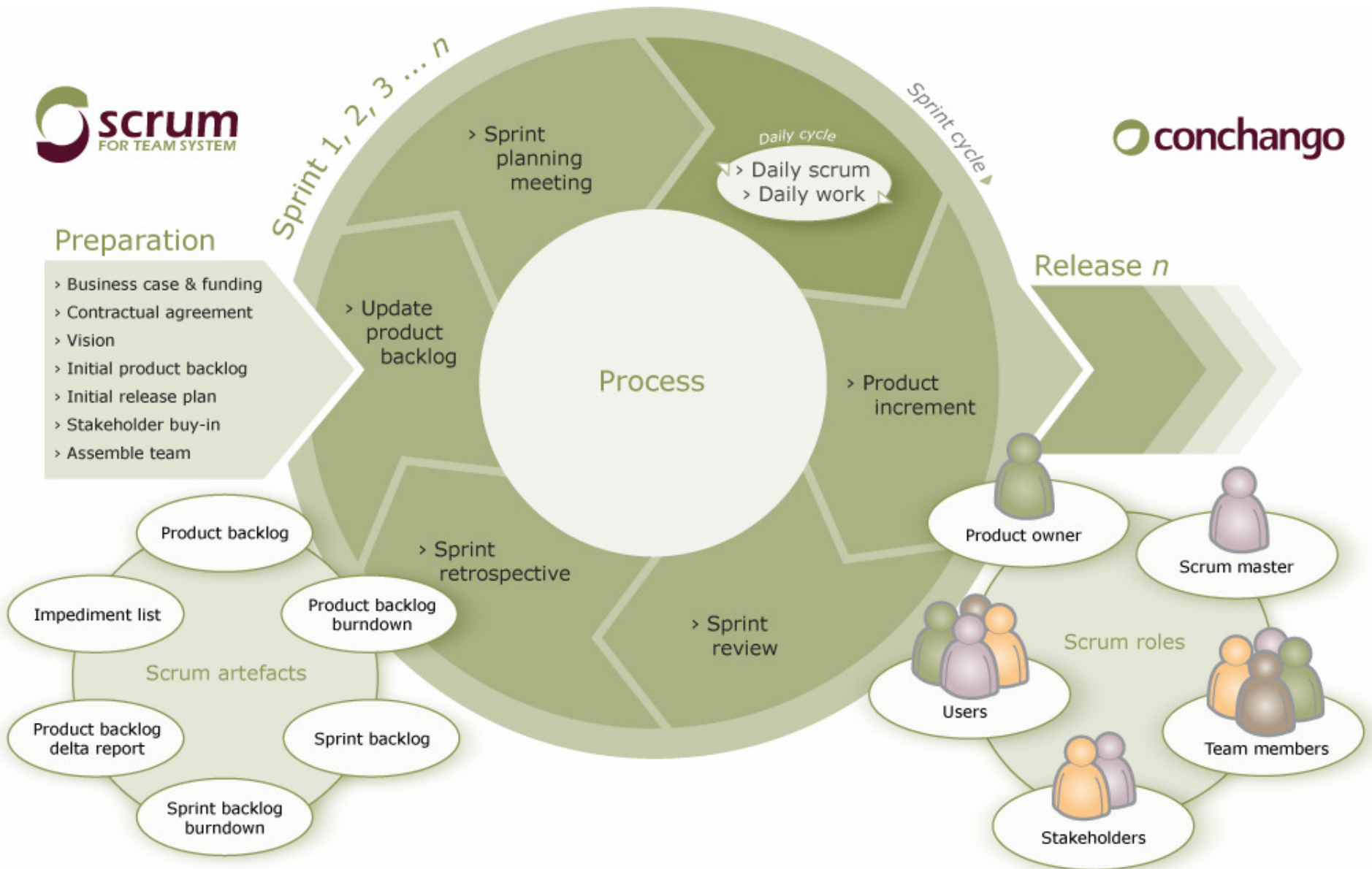- Refactor

Daisy Chain of Rules

- No detailed written requirements <-- no big up-front design.

- No big up-front design <-- constantly refactored.

- Constant refactoring <-- extensive unit testing.

- Unit tests <-- pair programmer.

- Pair programming <-- on-site customer.

- on-site customer <-- no detailed written requirements.
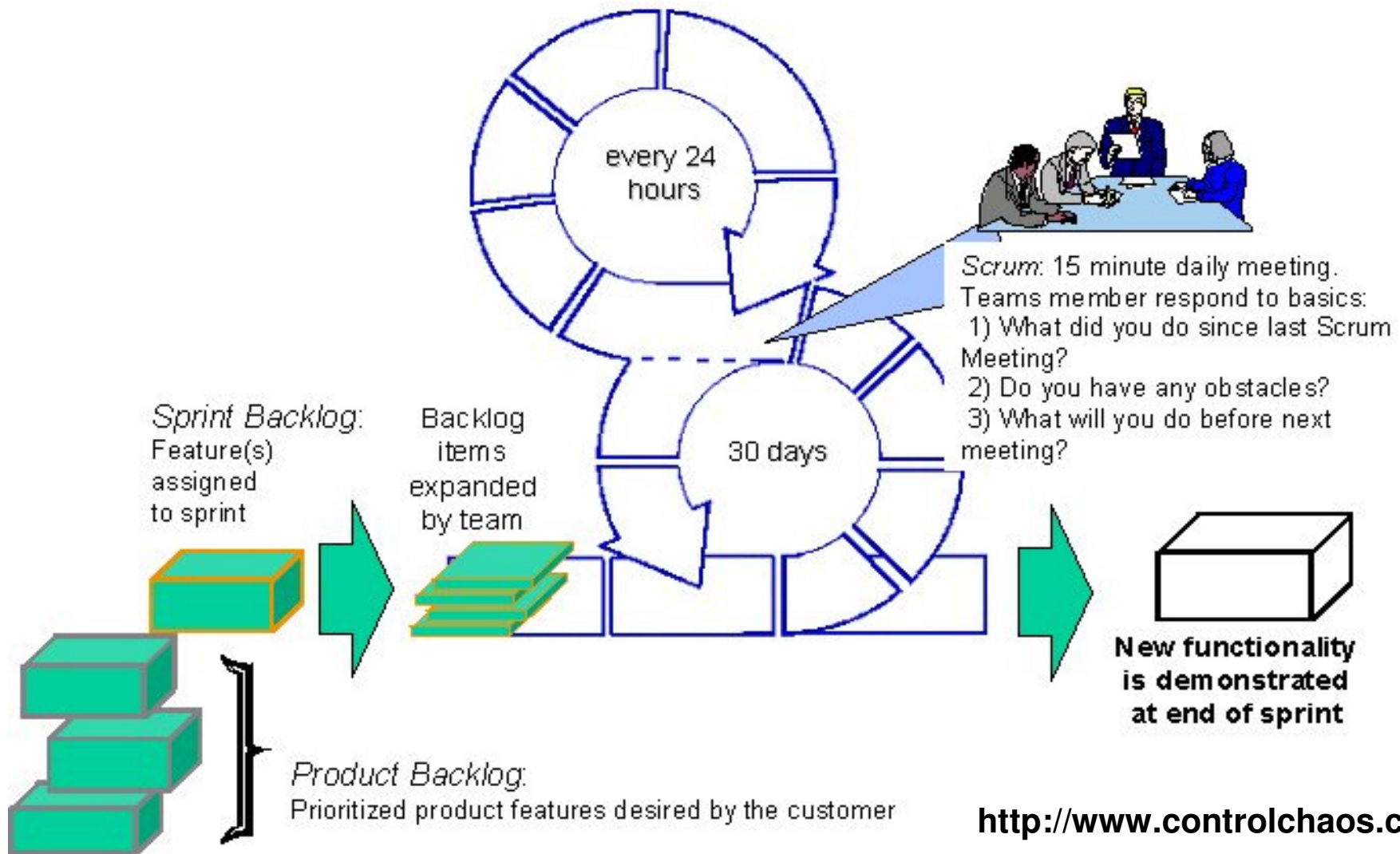
How XP Handles Problems

- Schedule Delays
  - Iterations
  - Short release cycles
- Project cancelled
  - Prioritize use case
  - Start with most important use cases
- System goes bad
  - Keep system always in deployable state
- Defects
  - Test Driven Development
- Business change
  - Continuous prioritization
- Business misunderstood
  - Customer always with team

Scrum

Scrum

every 24 hours

Scrum: 15 minute daily meeting. Teams member respond to basics:
1) What did you do since last Scrum Meeting?
2) Do you have any obstacles?
3) What will you do before next meeting?

Sprint Backlog: Feature(s) assigned to sprint

Backlog items expanded by team

30 days

New functionality is demonstrated at end of sprint

Product Backlog: Prioritized product features desired by the customer

http://www.controlchaos.com

Scrum Lifecycle

- **Pregame**
  - Planning - define system, Product Backlog
  - Architecture - high level design of system
- **Development**
  - Iterative cycles called sprints – plan, do, review
- **Postgame**
  - Operational deployment, documentation, training

Pre-Game	Mid-Game	Post-Game
Planning and High-Level Design	Develop, Wrap, Review, Adjust	Closure
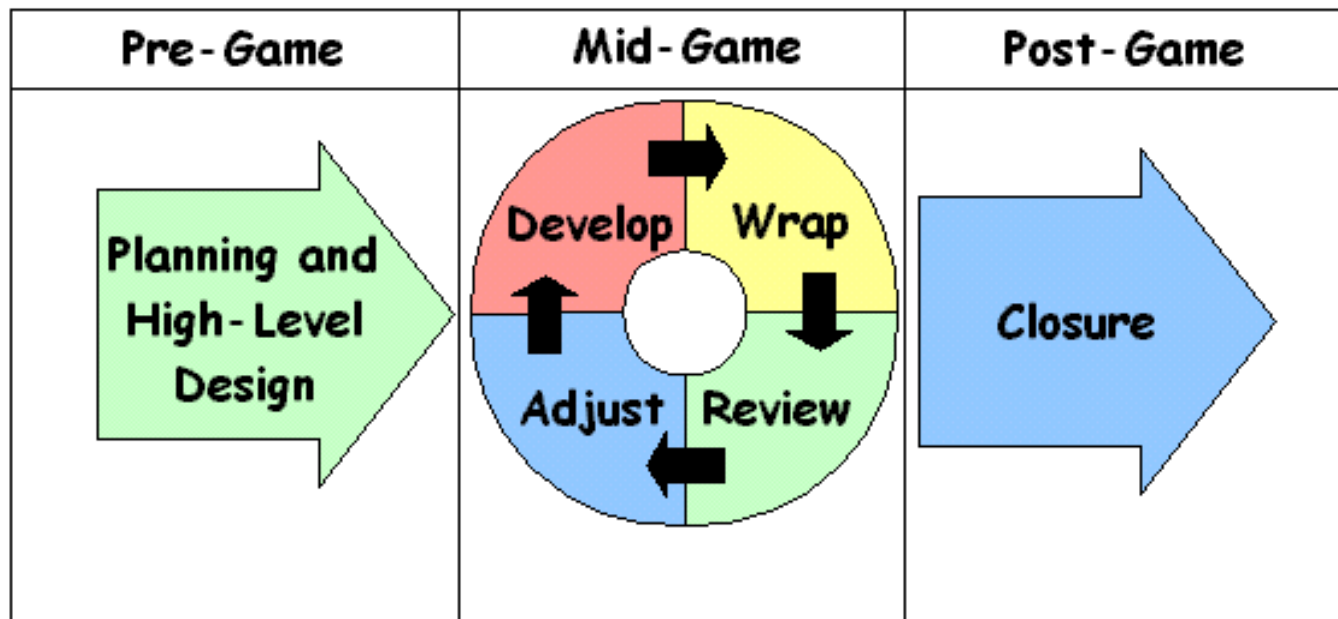
Scrum Roles

- **Scrum Team**
  - Project team that develops software
  - Determines sprint list
  - Free to organize as they see fit to achieve goals of each sprint
- **Scrum Master**
  - Responsible for Scrum Process.
  - Makes sure that project following rules and practices
  - Removes impediments
- **Product owner**
  - Creates and prioritizes Product Backlog
  - Chooses goals for next Sprint
  - Reviews system at the end of each Sprint
- **Customer**
  - Participates in Backlog items
- **Management**
  - Manpower allocation
  - Budgets
  - Billing

Scrum Practices

- **Pre-game Planning**
  - Product Backlog and Release Backlog
- **Sprint Planning**
  - Product Backlog, Release Backlog, Sprint Backlog
- **Sprint** - 30 day iteration
- **Effort Estimation** - iterative on Backlog items
- **Self-directed and self-organising team**
- **Daily Scrum meeting** - what we did, what we will do, and any problems
- **Chickens and Pigs**
- **Sprint Review Meeting** - present results of sprint, demo

Questions

Agile Software Development | Amit Goel

This has been fun…

- Thank You

- Feel free to drop me a line at
  amit@agoel.com

Backup Slides