Extreme Programming (XP)
Six Sigma
CMMI

How they can work together –
A JPMorgan Chase case study

Bob.Jarvis@chase.com
Stephen.P.Gristock@chase.com
Any statements made do not necessarily represent the views or opinions of JPMorgan Chase.
Agenda

Introductions
Exercise 1
Six Sigma Overview
XP Overview
CMMI Overview
Case Study – Six Sigma
  <Break>
Case Study – XP
Exercise 2
Case Study – CMMI
Lessons Learned
Parting Thoughts
Q&A
Introductions

Who Are We?

Why Did We Try XP?
Who Are We?

Steve Gristock
- CMMI Lead Appraiser & Instructor
- Proven Process Improvement Leader and Consultant

Bob Jarvis
- Six Sigma Coach
- Development Manager
JPMorgan Chase

Retail Financial Services
  - Home / Auto / Consumer / Small Business

Card Services

Investment Bank

Commercial Banking

Asset & Wealth Management

Treasury & Security Services
  - Treasury / Investor / Institutional Trust Services

Corporate
  - Private Equity / Treasury
Why Did We Try XP?

Typical Environment

- Project estimate accuracy
- Business – Technology working relationship
- Defect levels
- Overtime

Improvement Desired

- Better
- Cheaper
- Faster
- Work – Life Balance
Exercise 1: XP/CMMI: SURVIVOR!

- Can you survive the tribulations of the Six Sigma/CMMI/XP Survivor contest?
- Will you wail in anger and gnash your teeth if you’re voted off the island?
- Or- will you rise to the occasion and become Process Queen/King for the day?
- Do you care?
- Let’s play... and find out!
Six Sigma Overview

A Very Brief Overview
What is Six Sigma?

It’s an approach to managing a business
  ➢ Focus on clients, facts, measurement

It’s a process improvement methodology
  ➢ Improve existing processes
  ➢ Build new processes

It’s a calculation
  ➢ Allows us to measure quality consistently
Who’s Using Six Sigma?

Publicly traded companies that strategically highlight quality (Six Sigma / Baldrige Quality award winning companies) outperformed the S&P 500 by 4.8 to 1.¹

Key Drivers

Voice of the Customer (VOC)
- Critical to Quality (CTQs)
- CTQ Measures
- Voice of the ...
  - Business
  - Employee

Statistical Tools
- Analyze current state
- Verify results

Tollgates
- At every phase
## Terminology

<table>
<thead>
<tr>
<th>Process Improvement (DMAIC)</th>
<th>Process Design (DFSS – DMADVE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>➢ Define</td>
<td>➢ Define</td>
</tr>
<tr>
<td>➢ Measure</td>
<td>➢ Assess</td>
</tr>
<tr>
<td>➢ Analyze</td>
<td>➢ Select</td>
</tr>
<tr>
<td>➢ Improve</td>
<td>➢ Design</td>
</tr>
<tr>
<td>➢ Implement</td>
<td>➢ Implement</td>
</tr>
<tr>
<td>➢ Control</td>
<td>➢ Control</td>
</tr>
</tbody>
</table>
Process Synergy and Transition Points

---

**Define**

- Does a process / service / product exist?

---

**Measure** → **Analyze**

- Is current process capable of more?
  - Yes → **Improve**
  - No → **Assess** → **Select** → **Design** → **Implement** → **Control**

---

**Design**

- Is improvement a new process / service / product?
  - Yes → **Design for Six Sigma**
  - No → **New Project Idea**

---

**DMAIIC**

*Process Improvement*

---

**New Project Idea**
A DMAIIC Overview

Client-driven, consistent, metrics focused, results oriented

Define
- What are we trying to achieve?
- Who are the people to involve?
- Who has the knowledge required?

Improve
- What should we change to make improvements?
- How will we achieve this?
- Who’s ‘buy-in’ do we need?
- What should the controls be?

Measure
- Who are the clients?
- What are the current processes?
- How are we currently performing for our clients?
- How are we currently performing for our shareholders / employees?

Analyze
- Where are the problems with our current performance?
- What are the root causes?
- What are some quick hits for immediate improvement?

Implement
- Who are the clients?
- What are the current processes?
- How are we currently performing for our clients?
- How are we currently performing for our shareholders / employees?

Control
- Are the improvements being sustained?
- Are we continually measuring our performance against client expectations?
- Did we capture the learnings from this project?

- Is implementation on track?
- Are the controls in place?

- What are the clients?
- What are the current processes?
- How are we currently performing for our clients?
- How are we currently performing for our shareholders / employees?

- Where are the problems with our current performance?
- What are the root causes?
- What are some quick hits for immediate improvement?

- What should we change to make improvements?
- How will we achieve this?
- Who’s ‘buy-in’ do we need?
- What should the controls be?
XP Overview

WE'RE GOING TO TRY SOMETHING CALLED EXTREME PROGRAMMING.

FIRST, PICK A PARTNER. THE TWO OF YOU WILL WORK AT ONE COMPUTER FOR FORTY HOURS A WEEK.

THE NEW SYSTEM IS A MINUTE OLD AND I ALREADY HATE EVERYONE.

Copyright © 2003 United Feature Syndicate, Inc.
XP Context

What is Agile?

- An adaptive approach to solving business problems that focuses on communication, collaboration, delivery and change.
- “Outside the room.”

What is Extreme Programming?

- One of several agile methods.
- An innovative, deliberate and disciplined approach to software development.
- Developers, QA and Business in the same room (where applicable)
- “Inside the room.”
The Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.
What are the characteristics of an Agile process? An agile process …

… seeks to satisfy the customer through early and continuous delivery of valuable software as its highest priority.

… welcomes changing requirements, even late in development. Agile harnesses change for the customer’s competitive advantage.

… delivers working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter time scale.

… requires that business people and developers work together daily throughout the project.

… builds around motivated individuals. Give them the environment and support they need, and trust them to get the job done.

… promotes face-to-face conversation as the most efficient and effective method of conveying information to and within a development team.
An agile process …

… uses **working software** as the primary measure of progress.

… promotes **sustainable development**. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

… pays continuous attention to **technical excellence** and leverages good design to enhance agility.

… **demands simplicity**--the art of maximizing the amount of work not done.

… relies on **self-organizing teams** to generate the best architectures, requirements, and designs.

… asks the team to **reflect at regular intervals** on how to become more effective, then tune and adjust its behavior accordingly.
XP Values, Principles and Practices

“Anyone can make the simple complicated.

Creativity is making the complicated simple.”

Charles Mingus
Why “Extreme”?  

XP is a highly disciplined approach to software development that places quality at its core, and takes quality practices to the “extreme”:  

- **Testing**  
  - Failed unit tests = entry criteria for coding  
  - Unit tests = 100%  

- **Peer reviews**  
  - Pair programming  

- **Customer involvement**  
  - On-site, daily  
  - Customer-driven iteration content
Why “Extreme” (cont.)

- **Component integration**
  - Often / Continuous

- **Time to market**
  - Small releases

- **Refactoring**
  - Continual
  - Collective code ownership

- **Simplicity**
  - “The simplest thing that could possibly work”
XP – Values

- Communication
- Feedback
- Simplicity
- Courage
- Respect (new)
XP – Basic Principles

- Rapid feedback
- Assume simplicity
- Incremental change
- Embracing change
- Quality work
## XP Practices

XP is expressed through ~13 key disciplines (practices):

<table>
<thead>
<tr>
<th>Practice</th>
<th>Addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Planning Game</td>
<td>Priority, sequence, scope</td>
</tr>
<tr>
<td>Small releases</td>
<td>Time to market</td>
</tr>
<tr>
<td>Metaphor</td>
<td>Design context</td>
</tr>
<tr>
<td>Simple design</td>
<td>Incremental value delivery</td>
</tr>
<tr>
<td>Testing</td>
<td>Quality &quot;baked in&quot;</td>
</tr>
<tr>
<td>Refactoring</td>
<td>Code quality</td>
</tr>
<tr>
<td>Pair programming</td>
<td>Peer reviews, cross training</td>
</tr>
<tr>
<td>Collective ownership</td>
<td>Team culture</td>
</tr>
<tr>
<td>Continuous integration</td>
<td>Iterative build &amp; test</td>
</tr>
<tr>
<td>Sustainable pace</td>
<td>Work / life balance</td>
</tr>
<tr>
<td>On-site customer</td>
<td>Immediate feedback</td>
</tr>
<tr>
<td>Coding standards</td>
<td>Code quality</td>
</tr>
<tr>
<td>Whole Team</td>
<td>Teamwork</td>
</tr>
</tbody>
</table>
Practice “Rings”

Key
- Blue Ring: Developer practices
- Green Ring: Development team practices
- Red Ring: Entire team practices
XP – Global Presence

Source:
extremeprogramming@yahoogroups.com
“Simple, clear purpose and principles give rise to complex, intelligent behavior.”

“Complex rules and regulations give rise to simple, stupid behavior.”

Dee Hock
Founder and CEO emeritus, Visa International
When in doubt...

ACT STUPID!
Process Improvement & CMMI: Overview
Dealing With Process

- Process exists whether we acknowledge it or not. The only question is- do we take a structured and systematic approach to managing it, or do we allow it to develop organically?
## CMMI Process Areas

<table>
<thead>
<tr>
<th>CATEGORY</th>
<th>PROCESS AREA</th>
</tr>
</thead>
</table>
| Process Management | • Organizational Process Focus  
|                  | • Organizational Process Definition  
|                  | • Organizational Training  
|                  | • Organizational Process Performance  
|                  | • Organizational Innovation and Deployment                                  |
| Project Management | • Project Planning  
|                  | • Project Monitoring and Control  
|                  | • Supplier Agreement Management  
|                  | • Integrated Project Management  
|                  | • Risk Management  
|                  | • Quantitative Project Management                                           |
| Engineering     | • Requirements Management  
|                  | • Requirements Development  
|                  | • Technical Solution  
|                  | • Product Integration  
|                  | • Verification  
|                  | • Validation                                                               |
| Support         | • Configuration Management  
|                  | • Process and Product Quality Assurance  
|                  | • Measurement and Analysis  
|                  | • Causal Analysis & Resolution  
|                  | • Decision Analysis and Resolution                                          |
CMMI Representations

Staged
- Goals-Process Areas-Practices
- PA’s pre-selected
- Maturity levels (1-5)

Continuous
- Goals-Process Areas-Practices
- Select PA’s
- Capability levels within PA’s (0-5)
Staged CMMI Structure:

1: Initial
2: Project
3: Org
4: Metrics
5: Optimal
CMMI Continuous Structure:

0: Incomplete
1: Performed
2: Managed
3: Defined
4: Quantative
5: Optimizing

GOALS

PRACTICES

SUBPRACTICES

PROCESS AREAS

CAPABILITY LEVEL

GENERIC

SPECIFIC
CMMI & eXtreme Programming: Synergies
Conflicting Perspectives?

CMMI
- CMMI is an interpretive model
- At a Macro level, CMMI provides a framework for developing an end-to-end perspective for product development
- At a Micro level, CMMI provides process and practice solutions for controlling work

XP
- XP is a more specific set of prescribed methods
- XP provides the process and techniques required to deliver a collaboratively developed set of solutions in rapid succession
- XP is (necessarily) development-centric
CMMI & XP: The Stupid Seven

Misconceptions

- CMMI is too bureaucratic to coexist with XP
  - They’re definitely compatible if CMMI is interpreted and deployed appropriately
- CMMI requires a linear approach to software development
  - Use of CMMI is absolutely conducive to iterative development
- CMMI is only suitable for large organizations and projects
  - If scaled properly, CMMI may be deployed in large or small enterprises
- CMMI is incompatible with a collaborative development approach
  - CMMI is inherently compatible with integrated product development
- CMMI is a method and/or a standard
  - CMMI is a model, it should be used as a framework
- XP requires no documentation
  - Effective XP requires minimal, but consistent, documentation
- Agile development is creative and open with little or no real structure required
  - XP is a highly structured and disciplined method
JPMorgan Chase Case Study

Six Sigma Findings
XP Implementation
Results
Lofty Goals

Better
- Fewer defects

Cheaper
- Reduce project effort

Faster
- Reduce project duration

Quality of Life
- Enjoy work life better
- Do less of it
Executive Sponsorship

Business
- SVP – Internet Channel
- Senior Product Manager

Technology
- SVP / CTO – Regional Bank
- CTO – Internet Technology
Six Sigma Findings
### VOC - Business

For business: better, on-time delivery of agreed functionality (stories) are most important.

<table>
<thead>
<tr>
<th>#</th>
<th>Wtd</th>
<th>Voice of the Customer (VOC)</th>
<th>Critical to Quality (CTQ)</th>
<th>CTQ Measures</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.54</td>
<td>Quality product</td>
<td>Minimum defects</td>
<td># defects found in QA / UAT / production per unit of functionality</td>
</tr>
<tr>
<td>2</td>
<td>2.26</td>
<td>On-time delivery</td>
<td>All agreed stories delivered</td>
<td>% stories delivered for each iteration</td>
</tr>
<tr>
<td>3</td>
<td>3.52</td>
<td>All scoped functionality delivered</td>
<td>All committed iteration stories delivered</td>
<td>% stories delivered for each iteration</td>
</tr>
<tr>
<td>4</td>
<td>4.42</td>
<td>Faster time to market</td>
<td>Reduce time from story delivery to production</td>
<td># days / unit of functionality</td>
</tr>
<tr>
<td>5</td>
<td>6.04</td>
<td>Sound architecture</td>
<td>Applications are scalable, secure</td>
<td># hours of technology-driven rework</td>
</tr>
<tr>
<td>6</td>
<td>6.40</td>
<td>On budget</td>
<td>No cost overruns</td>
<td>$ variance</td>
</tr>
<tr>
<td>7</td>
<td>6.58</td>
<td>Accurate project scoping</td>
<td>All committed stories included in release</td>
<td># committed stories not included</td>
</tr>
<tr>
<td>8</td>
<td>6.76</td>
<td>Technical input on alternatives</td>
<td>Business understand technical trade-offs that may impact their decisions</td>
<td># unapproved technical / infrastructure stories requested by development</td>
</tr>
<tr>
<td>9</td>
<td>7.12</td>
<td>Business understands about technology / infrastructure / application limits</td>
<td>Informed business decisions are made</td>
<td># hours of technology-driven rework</td>
</tr>
<tr>
<td>10</td>
<td>7.66</td>
<td>Technology works within the business structure</td>
<td>Business can ensure their other touch-points are included as needed</td>
<td># hours waiting for business dependencies</td>
</tr>
<tr>
<td>11</td>
<td>8.20</td>
<td>Development activities fit in business resource constraints</td>
<td>Eliminate redundant documents / activities Decrease distractions (bus &amp; tech)</td>
<td># hours spent on redundant docs # hours / week distractions</td>
</tr>
</tbody>
</table>
CTQ Data

Top CTQs / CTQ Measures from VOC / VOB / VOE were combined to eliminate overlap (particularly around defect measures).

- **Committed Features**
  - % stories delivered

- **Defects**
  - # total defects / unit of functionality
  - # TRs related to requirements mis-match (WAD)

- **Costs**
  - $ / unit of functionality
  - # XP resources – deployment

- **Duration**
  - # days duration / unit of functionality

- **Miscellaneous**
  - # missed communication plan events
  - # manual steps - deployment
High quality and delivery of committed functionality (on time delivery) are top priority.

### Critical to Quality (CTQs)

<table>
<thead>
<tr>
<th>CTQ Measures</th>
<th>H</th>
<th>M</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimize defects</td>
<td>H</td>
<td>H</td>
<td></td>
</tr>
<tr>
<td>Deliver project success criteria</td>
<td>H</td>
<td>M</td>
<td>L</td>
</tr>
<tr>
<td>Committed stories delivered on time</td>
<td>M</td>
<td>H</td>
<td>L</td>
</tr>
<tr>
<td>Correct code delivered</td>
<td>M</td>
<td>M</td>
<td>L</td>
</tr>
<tr>
<td>Shorten duration - story to production</td>
<td></td>
<td>H</td>
<td>L</td>
</tr>
<tr>
<td>Maximize development ROI / effort</td>
<td>H</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Communication plan executed</td>
<td>H</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Symbol Meaning Score

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>High</td>
<td>9</td>
</tr>
<tr>
<td>M</td>
<td>Medium</td>
<td>3</td>
</tr>
<tr>
<td>L</td>
<td>Low</td>
<td>1</td>
</tr>
</tbody>
</table>

#### CTQ Importance = 6 - VOC Score (rank)

- Importance: 100 - 82 - 54 - 32 - 18 - 12 - 6 - 2
- Target: 0 - 100 - 0 - * - * - 0 - 2 - 1
- Specs: * - 95 - 100 - * - * - 0 - 1 - 0
- Key: "*" - Still to be defined

UOF - Unit of Functionality (size metric)
CTQ Importance = 6 - VOC Score (rank)
Break
JPMorgan Chase
Case Study (continued)

XP Implementation
High Level Process

Envision Phase
- Business vision
- Business case
- Road map
- Project charter

Speculate Phase
- Shared understanding
- Prioritized stories
- Story estimates
- Initial iteration plan

Iterate Phase
- Story details / tasks
- Test strategy
- Unit test / code
- QA / User sign-off

Release Phase
- Package
- Distribute
- Feedback response

Continuous Activities
- Project status
- Updated financials
- Communication plan
Anatomy of an Iteration

**Envision**

- Business Planning
  - Create business vision
  - Develop business case
  - Write high-level stories

**Speculate**

- Iteration Planning
  - Re-prioritized stories
  - Write story details
  - Prepare test scenarios
  - Identify technology spikes
  - Identify major refactoring

**Iterate**

- During Iteration
  - Update progress daily
  - Use test-first development
  - Pair program
  - Continuously integrate
  - Obtain QA (internal) sign-off
  - Obtain user sign-off

- Iteration Kick-Off
  - Determine availability
  - Identify story tasks
  - Update story estimates
  - Confirm plan

- Iteration Close
  - Confirm accomplishments
  - Discuss lessons learned
  - Update velocity

**Monitor**

- Release Planning
  - Create shared understanding
  - Prioritize stories
  - Estimate stories
  - Identify dependencies

- QA

- UAT

- Additional Iterations Based on:
  - Required content
  - Release date
## Parallel Activities

<table>
<thead>
<tr>
<th>Week n-6</th>
<th>Week n-4</th>
<th>Week n-2</th>
<th>Week n</th>
<th>Week n+2</th>
<th>Week n+4</th>
<th>Week n+6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Envision</td>
<td>Speculate</td>
<td>Iterate</td>
<td>Plan</td>
<td>Develop</td>
<td>Plan</td>
<td>Develop</td>
</tr>
<tr>
<td>QA / UAT</td>
<td>Monitor</td>
<td>Plan</td>
<td>Bus</td>
<td>IT</td>
<td>Bus</td>
<td>IT</td>
</tr>
</tbody>
</table>

Week n:
- Envision
- Speculate
- Plan
- Develop
- Plan
- Develop
- Monitor
- QA / UAT
- Plan
- Develop
XP Stories

EXTREME PROGRAMMING

I CAN'T GIVE YOU ALL OF THESE FEATURES IN THE FIRST VERSION.

AND EACH FEATURE NEEDS TO HAVE WHAT WE CALL A "USER STORY."

OKAY, HERE'S A STORY: YOU GIVE ME ALL OF MY FEATURES OR I'LL RUIN YOUR LIFE.

Copyright © 2003 United Feature Syndicate, Inc.
The Story

The story is a unit of functionality in an XP project. We demonstrate progress by delivering tested, integrated code that implements a story.

Story Evolution

- **Business Vision**
  - Long-term functionality view (6-18 months)

- **High-level Stories**
  - Functionality that delivers value
  - Small enough to estimate
  - Prioritized

- **Story Details**
  - “Just enough” detail
  - Use cases work well
  - Includes high-level test scenarios
  - Updated to reflect reality
Story Tracking

Future

Ready

Active

User Ready

QA Ready

Done

Release Ready

External QA
### Revised Direct Deposit Landing Page

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Owner</th>
<th>P</th>
<th>H</th>
<th>C</th>
<th>D</th>
<th>B</th>
<th>TR #</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1-2004-SS</td>
<td>Robert</td>
<td>2</td>
<td>13</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Iteration Development | 121 | Self-Service
User Sign-Off
Wiki Wiki

(Hawaiian for quick quick)
Repository Contents

- Wiki Overview
- Iteration Management
  - Time Tracker
  - Current Iteration
  - Iteration Details
  - JOE Awards
- Development
  - Test Coverage
  - Interesting Items
- QA
  - Functional Testing Rules
  - CFT Knowledge Transfer
    - XP QA Automation Matrix

Front Page
Results

“In God we trust.

All others must provide data.”

W. Edwards Deming
Results - Metrics

Defects
- Total: Include all severities
- Critical: Only the highest severity
- Working as Designed: Points to business / technology disconnect

Effort & Duration
- Cost & calendar time

Size - QA test cases
- Best size metric

Quality of Life
- Business: 81% better / much better
- Technology: 77% better / much better
Satisfaction Categories

- Overall job satisfaction
- Work / life balance
- Quality of work environment
- Teamwork
- Relationship with bus / tech counterpart
- Quality of tools
- Feel valued by the company
- Effectiveness at your job
- Level of accomplishment
Metrics – Defects

Total Defects
63% reduction

Critical Defects
79% reduction

Working As Designed
38% reduction
Metrics – Effort / Duration

Duration (Days)

- Non-XP A
- Non-XP B
- Non-XP C
- XP 1
- XP 2
- XP 3
- XP 4

Effort (Hours)

- Non-XP A
- Non-XP B
- Non-XP C
- XP 1
- XP 2
- XP 3
- XP 4

44% reduction

47% reduction
Other XP Metrics

How We Measure Ourselves
Metrics Categories

Release Level
- Defects
  - Total
  - Critical
  - Working as Designed (WAD)
- Effort / Duration

Iteration Level
- Velocity
- Stories Delivered

Daily
- IDH Delivered
- IDH Remaining
Metrics – Iteration

Velocity

Stories Delivered

Iteration

Actual
Moving Average

Planned
Delivered
% Moving Avg.

Iteration

Iteration

0.60
0.50
0.40
0.30
0.20
0.10
0.00

2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

23 45 678 9 10 11 12 13 14 15 16 17 18 19

0%
25%
50%
75%
100%
125%
150%

0%
25%
50%
75%
100%
125%
150%

Exercise 2: XP/CMMI Cliché Combat

- Our goal is to invite, and capture, as many misinformed “pearls of wisdom” related to Six Sigma/CMMI/XP.
- Whether they are direct quotes or simple anecdotes, we invite you to approach the mic and share.
- In order to get the “juices flowing” and the “ball rolling”, here are some of our favorites…
XP/CMMI
JPMorgan Chase
Case Study (continued)

The CMMI Perspective
JPM Strategy

Roadmap:

- Baseline Assessment
- Implement:
  - Address Gaps
  - Establish PAL
  - Deploy & Train
  - Upgrade
  - Interim Assessment
- Measure (Assess/metrics)
Performed a 2-day CMMI Class C (Mini Assessment), based upon very limited scope and sample set.

Indicator Scores:
- 1-3: Weak
- 3-5: Progressing
- 6-7: CL 2
- 8-10: CL 3
The Baseline Assessment indicated that the XP program was on the cusp of satisfying CMMI Capability Level 2 across the in-scope Process Areas (PA).

By leveraging infrastructure and assets from the existing PI program, and because of the relatively fast XP cycle-times, it was feasible to set an aspirational goal of attaining a CMMI Capability Level (CL) 3 profile within 9 months.
Developing & Implementing The XP/CMMI Process

- Performed a series of workshops with the entire team to capture and define XP practices in standard SDLC format
- Integrated XP SDLC within organization’s online Process Asset Library
- Promoted awareness and conducted briefings and OJT across team
- Captured feedback and adjusted XP SDLC
- Performed an interim CMMI Class B Assessment
Organizational Process Library

- CMMI
- Policies
- Corporate PLC

**SDLCs**

- Project Initiation
- Fast Ref Processes
- Detailed Processes
- Assoc Templates

**XP SDLC**

**WIKI Repository**

**PPQA**

**PMO**

Tailoring based upon optional and mandatory criteria applied at each level.

Project Team Point Of Ref
Conclusions:

XP/CMMI Working In Concert!

- Deployment of CMMI can provide a framework for implementing a more robust XP-based method.
- The rapid cycle times associated with XP enable correspondingly quick development, piloting and deployment of a CMMI aligned process.
- CMMI contains a robust definition of engineering practices.
- Because of the emphasis on product development, CMMI provides a useful framework for engaging non-development groups (i.e. the business, operations, governance, support etc.).
- If deployed as part of an organizational initiative, CMMI can help promote awareness and propagate XP practices.
Lessons Learned

“The first step towards getting somewhere is to decide that you are not going to stay where you are.”

John Pierpont Morgan
Summary

- **Strategic Initiatives**
  - Six sigma excellent for introducing XP
  - XP highly compatible with CMM / CMMI

- **Organizational**
  - Throughput is closely tied to organizational agility
  - Barriers are stronger in minds than in reality
  - Success is directly tied to level of business / technology collaboration & availability

- **Project Preparation**
  - Infrastructure setup required before development starts

- **Business ready for collaborative planning**

- **Training**
  - Business & technology should be trained together

- **Consultants Speed Adoption**
  - Business
  - Technology
Challenges

- **Reduce Time-to-Value**
  - “Fast track” for project subset?

- **Team distractions**
  - “Distraction-free zone”

- **Reduce principle on current “debt”**
  - Automated UT / IT scripts for existing code base

- **Environment testing (OS / browsers)**
  - VMWare

- **Keep enthusiasm in check**
  - There’s much to learn

- **Decision-Making**
  - Empowered, risk-accepting

- **Managing Change**
  - Finding the balance
XP Alone

Does XP work? … Yes

Is XP optimized? … No
As more development partners become agile, both they and the XP team gain efficiency and effectiveness.

Net effect = increased throughput
Parting Thoughts
Complementary Approaches

Six Sigma
- Driven by business needs
- Disciplined implementation
- Results verified through metrics

Extreme Programming
- Better / cheaper / faster
- Improved quality of life

CMMI
- Recognized framework
- Lends legitimacy
Complementary Approaches

Six Sigma

Guides Implementation

Extreme Programming

Provides Process Framework

CMMI
Q&A

Audience Participation Encouraged

Where are you today?
Contact Info

Bob Jarvis
- Bob.Jarvis@chase.com

Steve Gristock
- Stephen.P.Gristock@chase.com