PowerWorld Development Process

August 23, 2011

Caroline Marzinzik

caroline@powerworld.com, 217-384-6330 ext. 12

Jamie Weber

weber@powerworld.com, 217-384-6330 ext. 13

PowerWorld Corporation

Topics

- Software Development Process that PowerWorld Corporation Uses
  - Extreme Programming
- PowerWorld Development Team
- Tools Used by PowerWorld
  - Borland (now called Embarcadero) Delphi
  - Code Management System
- Patch Versions
- How PowerWorld decides what new features to develop

Software Development Process: Waterfall Process

- What you learned in school was the "Waterfall Process"
  - http://en.wikipedia.org/wiki/Waterfall_model
  - Customer States Requirements
  - Design for Requirements
  - Implementation (coding)
  - Verification (testing)
  - Maintenance

What's wrong with the Waterfall Process

- Requirements are usually constantly changing
  - You end up writing software that solves the problem you *thought* you had instead of what you end up really having.
- Customer is only involved up front – doesn't allow them to guide changes in requirements
- Part of the design may be too difficult to implement requiring a reconsideration of the design
  - Or design may just be too costly
- If you wait until the end to test, then you might spend a huge amount of resources on software that has no hope of working
  - Incremental development builds confidence that the software design will actually work
  - Incremental development allows you to abandon a bad design early on

What PowerWorld Uses: Extreme Programming (XP)

- This is a real development process with years of history and corporate research behind it
  - http://www.extremeprogramming.org
  - http://en.wikipedia.org/wiki/Extreme_Programming
  - Process is possible because of object-oriented programming
    - software can be written and updated in pieces
- Extreme Programming Values
  - Communication
  - Simplicity
  - Feedback
  - Courage

Extreme Programming: Communication

- Communicate with the customer in all parts of the process
  - Developers must understand what the customer is trying to do, not just what the software does
- Developers also must communicate extensively with one another
  - More than one developer is familiar with each part of the software
    - More than one person can help with bug fixes and support
    - Employee turnover and retirements won't kill the software
  - Developers have a shared vision

Extreme Programming: Simplicity

- Implement software in the simplest manner initially
  - Might not be the optimal approach initially, but software can be improved later
  - Prevents you from wasting time optimizes parts of the software that are not important
  - Customer will communicate with feedback what needs to work better
  - Makes is so all developers can understand most of the software

Extreme Programming: Feedback

- Send software updates frequently to customer and get feedback on functionality as well as design
- Give feedback to the customer about estimates of how much time component tasks are going to take
- Feedback from the system by testing the software

Extreme Programming: Courage

- Have courage to *refactor* the code
  - Completely rewrite a section of code so that it can be better modified in the future
  - This goes hand-in-hand with Simplicity because you may need to rewrite some old code to improve performance
- Know when to throw-away your code
  - Just because you spent time on some code doesn't mean it's worth keeping

PowerWorld Software Development Team – 9 permanent people

- Tom Overbye, Ph.D. (1994 and before)
  - Company founder and a professor at University of Illinois
- Jamie Weber, Ph.D. (1997)
  - Director of Operations
- Kollin Patten, M.S.E.E. (1997)
  - Director of Engineering
- Kyle Johnson, B.S. C.S. (2003)
- Caroline Marzinzik, M.S.E.E. (2004)
- Matt Davis, Ph.D. (2009, part-time 2003)
- Angel Aquino-Lugo, Ph.D. (2010, part-time 2007)
- Thomas Nicol, M.S.E.E. / B.S. C.S. (2011)
- Kate Rogers, Ph.D. (2011, part-time since 2007)

PowerWorld Corporation Business and Consulting Team

- Mark Laufenberg, Ph.D. (July 1996)
  - President
- Kelley Wegeng (May 1999)
- Scott Dahman, M.B.A, M.S.E.E (August 2003)
  - Director of Business Development
- Santiago Grijalva, Ph.D. (May 2001 - 2009)
  - Left in August 2009 to be a tenured professor at Georgia Tech

PowerWorld Software Team Support

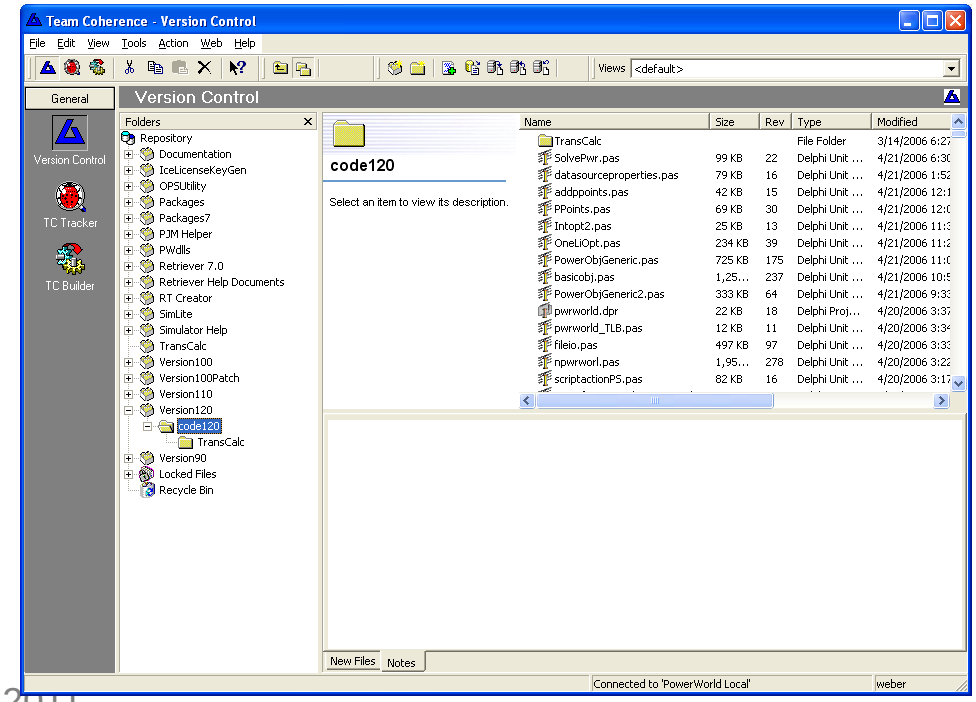
- All developers are involved in all parts of the software process
  - Write software
  - Provide email and phone support
  - Give training
  - Write help documentation
  - Communicate with customers regarding potential new features

Tools used by PowerWorld: Development Language

- Development Language
  - Embarcadero Delphi XE presently
    - Visual, Object-Oriented, windows development language
  - Most of code is written in-house
  - Make use of third-party, royalty-free software when appropriate
    - Toolbars and Menus
    - HTTP, HTTPS, FTP, FTPS Connections
    - Lock and Key Software
    - UCalc32 DLL (parses expressions)
    - Zip/Unzip DLLs (PowerWorld Project files)

Tools used by PowerWorld: Code Management System

- Team Coherence Version Manager
  - Made by Quality Software Components (QSC)
  - http://www.qsc.co.uk



Tools used by PowerWorld: Code Management System

- Allows all developers to keep their code synchronized
- In order to edit a file a developer must check it out
  - This locks in down so other developers cannot edit it
- When finished editing, code is checked back in
  - Code is always in a state that can be compiled
- Maintains the revision history of all source files
- Allows remote users to work on the same files

PowerWorld Patch Maintenance

- When a new version of the software is released we create a new “project” in the Team Coherence Software
  - This means that we have two sets of source code that must be maintained
  - Future development is performed on the new project
  - Bug fixes are made to both projects
  - Simple new features may be added to both projects
    - Allows us to get new features out quickly
    - Features which effect the file format can not be added.

Putting out a new patch version

- When a new patch version is necessary, a label is defined in the Team Coherence tool and the present state of the patch source code is marked with the date of the patch
- New patch versions are compiled
- Patches are placed on our website
- Website is updated to reflect the changes that have been made to the patch
- This occurs on the order of once a week

How PowerWorld decides what new features to develop

- Bug fixes come first
  - New development must wait for bug fixes to be complete
  - Bug fixes are sent out via the patch website
- Extreme Programming Rule – Customer Feedback
  - When a few customers request a simple change it will be made immediately
- For Larger software modifications
  - Market Need
    - Is there a big enough market to support the development costs of the software tool?
  - Customer cost-sharing and testing arrangement is typical
    - Historically, PowerWorld has between 2 and 4 cost-sharing arrangements going on at all times
    - Range of these projects is $1,000 - $400,000
    - Most projects are $20,000 - $40,000 (a couple man-months)

How PowerWorld decides what new features to develop

- Bug fixes come first
  - New development must wait for bug fixes to be complete
  - Bug fixes are sent out via the patch website
- Extreme Programming Rule – Customer Feedback
- When a few customers request a simple change it will be made immediately
- For Larger software modifications
  - Market Need
    - Is there a big enough market to support the development costs of the software tool?
  - Customer cost-sharing and testing arrangement is typical
    - Historically, PowerWorld has between 2 and 4 cost-sharing arrangements going on at all times
    - Most projects are $20,000 - $40,000 (1 or 2 person-months)
    - We do them as small as $1,000 though

What the Customer gets from a Cost-Sharing Arrangement

- Work with a company with a proven track-record of providing software that meets their customers needs and supports the software after completion
- Get to drive the development of an important software tool using outside software developers
  - More cost-effective than writing custom software
  - Customizations integrated in Simulator are maintained and supported by PowerWorld going forward
  - Software is tested by the PowerWorld User-Community as a whole
- Customers' staff is able to become intimately aware of the functionality of PowerWorld
- Minor requests from cost-sharing clients are typically added at no cost

What PowerWorld Gets From a Cost-Sharing Arrangement

- Extreme Programming Principles
  - Customer is intimately involved in designing the software needs
  - Customer contributes their own staff time to providing feedback and testing
  - PowerWorld staff learns tremendously from the customer interaction
- Defrays the development cost
  - Not looking to recoup all our costs
- Proves that the enhancement meets a market need

Brief History of Cost-Sharing Development

- Development of Visualization Software
  - NSF Small Business Innovative Research Grants (1997 – 2003)
  - Wisconsin Public Power (1997)
- Sensitivity Analysis Tools
  - American Public Power Administration (1998 – 1999)
  - California Energy Commission – Weighted TLR Sensitivities (2002)
- OPF and SCOPF Development
  - US DOE – EIA (1998 – 2002)
  - Tennessee Valley Authority (2001 – 2004)
  - Dynegy (2002)
  - NSF SBIR Grant for OPF enhancements
  - OPF Reserves, Midwest ISO (2007)
- PVQV, ATC, Contingency Analysis, Misc Development
  - Bonneville Power Administration (1999 – present)
  - NSF SBIR for PVQV tool
- SimAuto Development
  - Enron (2000)
- File Format Management (EPC, RAW, AUX)
  - Bonneville Power Administration (1999 – present)
  - Columbia Grid (2010)

Brief History of Cost-Sharing Development

- Geographic Information System Support, Line Impedance Calculator
  - Defense Intelligence Agency (DIA) (2004 – 2007)
- Retriever Development
  - Richmond Power and Light (1998)
  - American Public Power Administration (1998 – 1999)
  - Commonwealth Edison (1999)
  - Tennessee Valley Authority (2000 – present)
  - PJM Interconnection (2003 – 2005)
  - ISO – New England (2004 – 2007)
  - SPP (2005 – 2007)
  - NSF Small Business Innovative Research Grant (2003 – 2005)
- Integrated Topology Processing
  - ISO – New England (2004 – 2008)
  - TransGrid in Australia (2007 – present)
  - Bonneville Power Administration (2008 – present)
- Transient Stability
  - DIA Stability Data File Support (2007)
  - Glover, Sarma, Overbye Textbook (2002)
  - Bonneville Power Administration (2008 – 2010), ongoing interaction
- PowerWorld Trainer
  - TransGrid in Australia (2011)

Present On-Going Software Development Contracts

- Tennessee Valley Authority
  - Ongoing Retriever modifications to support full-topology models
- NERC
  - Situational Awareness for FERC, NERC and the Regions (SAFNR) project to build a real-time visualization of the entire country
- TransGrid
  - Ongoing work with Trainer
  - Automation of Retriever Model Creation
- Bonneville Power Administration
  - Ongoing full-topology model related features
  - State Estimation Project (Starting around October 2011)
  - Retriever Advanced Visualizations for Control Room and of PMU measurements (Starting around October 2011)

PowerWorld Corporation Development Process Summary

- Flexible Development
- All developers are involved in every facet of the process
  - Ensures that knowledge is not locked in one employee
  - Clients can talk to any of us and get help
- Customers drive the development
  - Customer feedback is vitally important to us
  - Call us: (217) 384-6330
  - Email us: support@powerworld.com