

*Designing for Easy Release
(a whistle stop tour...)*

Robert Cowham
robert@vaccaperna.co.uk
Vaccaperna Systems Ltd

Easy Releasing Contents

- F** Definitions and Common Problems
- F** Recommendations:
 - Planning
 - Process/Patterns
 - Practice
- F** Case Study
- F** Summary

Definitions and Common Problems

What is a Release?

- F** Getting a version of "the system" into the hands of the users
- F** Can you deliver "the system" as "a product"?
 - That works/that the users like/...
- F** System may be:
 - Executable/web page/...
 - Documentation/help
 - Related "backend" systems
 - Etc.

Releases - The Business Perspective

- F** They Take too long
 - Market opportunity
- F** Unpredictable
- F** Risk management
- F** Cost of support
 - Reworking "in the field"
- F** User perception

Other Release Problems

- F** Contents unplanned
- F** Contents unknown
- F** Distribution/installation
- F** Upgrade/downgrade
- F** Support/Helpdesk
 - What's "in the field"?

Planning

(c) 2004 Vaccaperna Systems Ltd

7

VACCAPERNA
SYSTEMS LIMITED

Lifecycle Impact of Releasing

- F** Requirements (market/features)
- F** Architecture
- F** Build (integration)
- F** Test
- F** Deliver/Install
- F** Maintain/upgrade

(c) 2004 Vaccaperna Systems Ltd

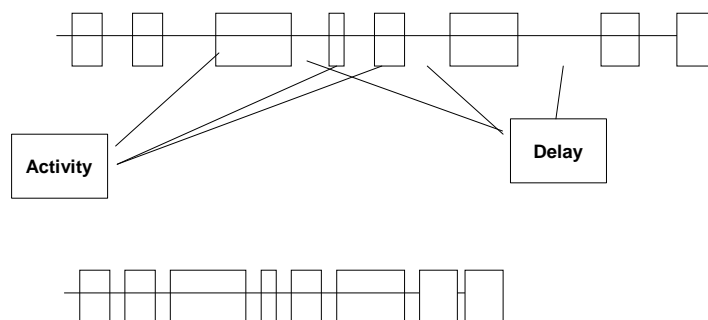
8

VACCAPERNA
SYSTEMS LIMITED

*Lean Development Principles
(Mary & Tom Poppendieck)*

- F** 65% of features are rarely or never used (Jim Johnson of the Standish Group)
- F** Streamline your process
- F** Minimum marketable feature sets
- F** Shorten Customer Feedback Loop
 - Requirements change

Value Stream Mapping



Architecture

F Often already defined

F Types include:

- Fat client
- Thin client/Browser based
- .NET/Java
- Embedded

F Windows based product

- .NET vs. COM?

Process/Patterns

F Some applicable patterns

Software Design Principles

- F** Correctness
- F** Modularity
- F** Cohesion
- F** Coupling
- F** Understandability
- F** Adaptability

SCM Patterns *(Berczuk & Appleton)*

- F** Task Level commit
- F** Integration Build
- F** Smoke Test/Regression Test
- F** Release Prep Codeline
 - Policy

Producing Release Notes

F Key indicator for process health

F List of changes

- Bugs fixed/features added
- Code changes made by developers

Traditional Release Notes

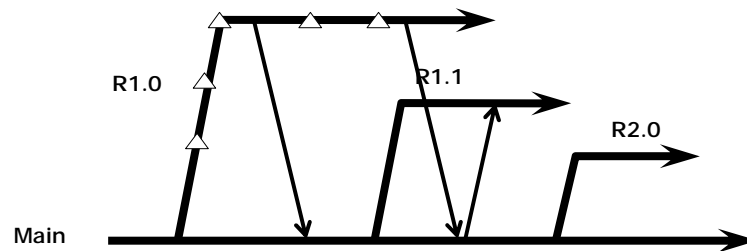
F My experience...

- Lastminute.com(mon) activity...
- Asking/pleading with developers
- Grubbing around in the code

Easier Release Notes

- F** Task driven development
 - Release contents planned
 - No code changes without associated task
- F** Workflow
- F** Automated link of task to code change (change set)
- F** Tool support to track associations across release lines (and report)

Track Change Propagation



System Environment

- F** Database
- F** Third Party Systems
- F** Live feeds

CM & Databases

- F** Key component of many systems
- F** CM often done as part of DB Design only - not CMed
- F** What to control?
 - DB Structure as SQL scripts
 - Table contents?
 - Configuration tables
 - Upgrade/downgrade scripts

DB Upgrade Issues

- F** New tables
- F** Change table structure
 - Need to convert existing data?
 - Add/change new columns (based on current info)
- F** Can scripts/procedures cope with already converted DB?
- F** Access to older format tables, e.g. transaction logs

Practice

makes perfect...

How to Release Better?

F Do it more frequently!

Agile Development

F Regular releases (short cycle)

F Continuous Integration

F Test driven development

- Unit testing
- Acceptance testing

F Code should always work!

- Policy

Automation for Frequent Releases

- F** Integration
- F** Build
- F** Test
- F** Documentation
 - Help/other docs
 - Release Notes
- F** Install
 - Upgrade

Windows Automation Issues

- F** IDEs not always easy to script for build
- F** Installation
 - DLL Hell
 - .NET XCopy Deployment

Testing

- F** Test driven development => easier to test
- F** What are the links to the external world?
 - Can we only test in live environment?
 - NATS!

Case Study

- F** Large financial system
- F** 3 releases
- F** 15 clients
- F** 1 single development workspace
- F** Getting a bug fix required taking a large number of new features
 - Could take weeks to stabilise a release

Conclusion

Easy Releasing Summary

- F** Plan it
- F** Process/Patterns
- F** Practice
 - Iterative/agile methodologies

And finally...



Copyright © 1995 United Feature Syndicate, Inc.
Redistribution in whole or in part prohibited