

Software Configuration Management



Lecture 5: Release Management & Roadmap & Variability

René Krikhaar

Questions and Remarks?



- ◆ Lecture 4, 3, 2 and/or 1 ?
- ◆ Student Presentations ?
- ◆ Papers or Literature ?
- ◆ Any other remarks ?

Lecture 1b:

Introduction SCM



- ◆ What are the four (five) components in SCM?
- ◆ Give a definition of Software Configuration Management in your own words
- ◆ Which functions belong to SCM?
- ◆ Give some examples of configuration items (CI's)

Lecture 2: Version & Build Management



- ◆ Why do we need versions of CI's?
- ◆ What (user) actions are defined on CI's to realize versioning?
- ◆ What is Build Management?
- ◆ What is the relation between Version Management and Build Management?

Lecture 3: Software Merging & Awareness



- ◆ What is software merging?
- ◆ Why do we need software merging?
- ◆ What kinds of software merging tools exist?
- ◆ What is awareness?
- ◆ Why do we need awareness?

Lecture 4: Branching Strategies etc.



- ◆ Why do we need software branching ?
- ◆ How to implement software branching ?
- ◆ What is Version Space & Product Space ?
- ◆ Which metrics influence performance of SCM ?

Lecture Objectives



- ◆ Learn about Release Management (theory and practice)
- ◆ Learn about the future of SCM
- ◆ Learn about Variability Management

Release Management



- ◆ A case study of the Release Management of a Health Care Information Systems
Gerco Ballintijn, ICSM 2005
- ◆ Group PR5:
 - ◆ Qing Gu
 - ◆ Wim Koomen

SCM Roadmap



- ◆ Software Configuration Management: A roadmap

Jacky Estublier, ICSE 2001

- ◆ Group PR4:
 - ◆ Edwin de Vries
 - ◆ Steven Heijtel

Variability Management



- ◆ Representing variability in a family of MRI scanners

M. Jaring, R. Krikhaar, J. Bosch

- ◆ Group PR6:
 - ◆ Agung Pramono
 - ◆ Guido van Stormbroek
 - ◆ Johan V. Vries
 - ◆ Jan Kubr

Release Management



- ◆ Making a product available to its intended customers
 - ◆ External Release (Customers)
 - ◆ Internal Release (Developers)
- ◆ Identification of a Product Release by a “Release Number”

Customer Release Classifications – 1 –



- ◆ Alpha Release
 - ◆ When? Significant increase of functionality
 - ◆ Purpose? Find defects and evaluate
 - ◆ Outcome? Bug repairs & New features
- ◆ Beta Release
 - ◆ What? Find defects in product
 - ◆ Outcome? Bug repairs (no new features)
- ◆ Final Release
 - ◆ Customer release

Customer Release Classifications – 2 –



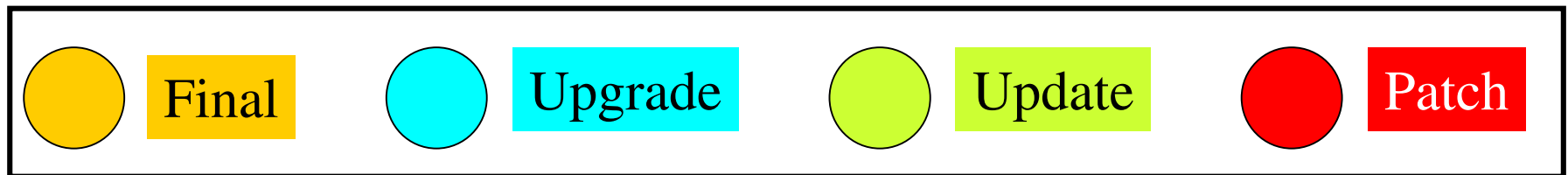
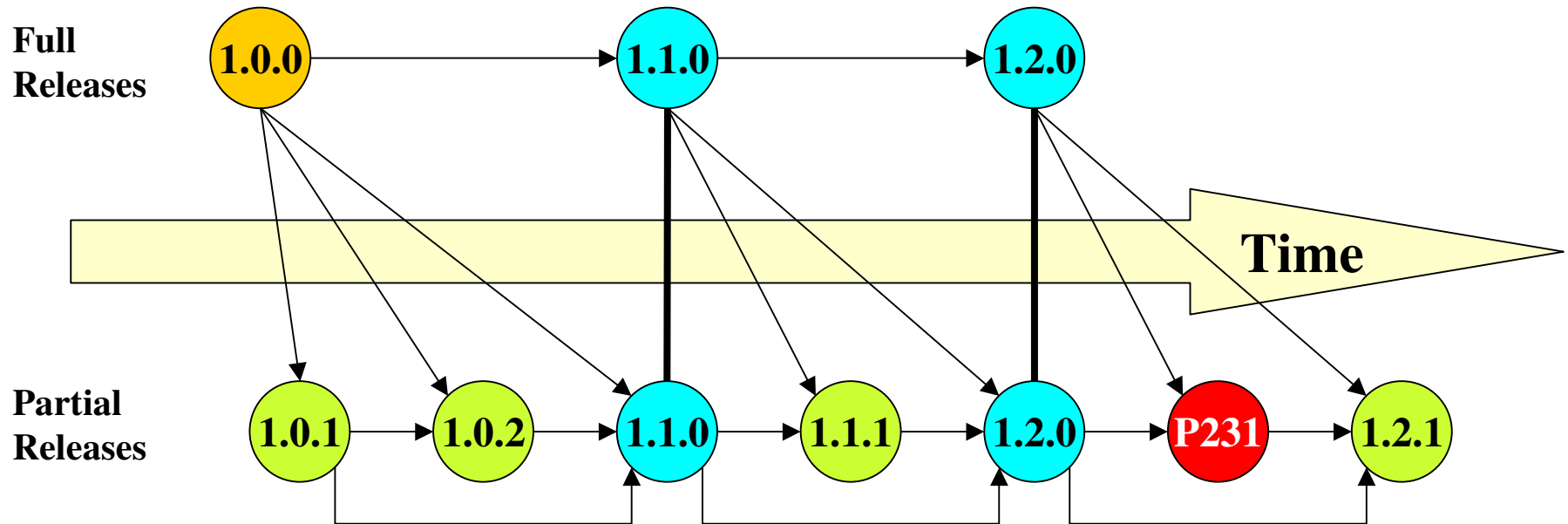
- ◆ Update Release
 - ◆ Fixing defects (update)
- ◆ Upgrade Release
 - ◆ Add (small) features (upgrade)
- ◆ Patches (Emergency Fixes)
 - ◆ Emergency (quick solution; can't wait)
 - ◆ Solve customer's difficulties

Release Numbering

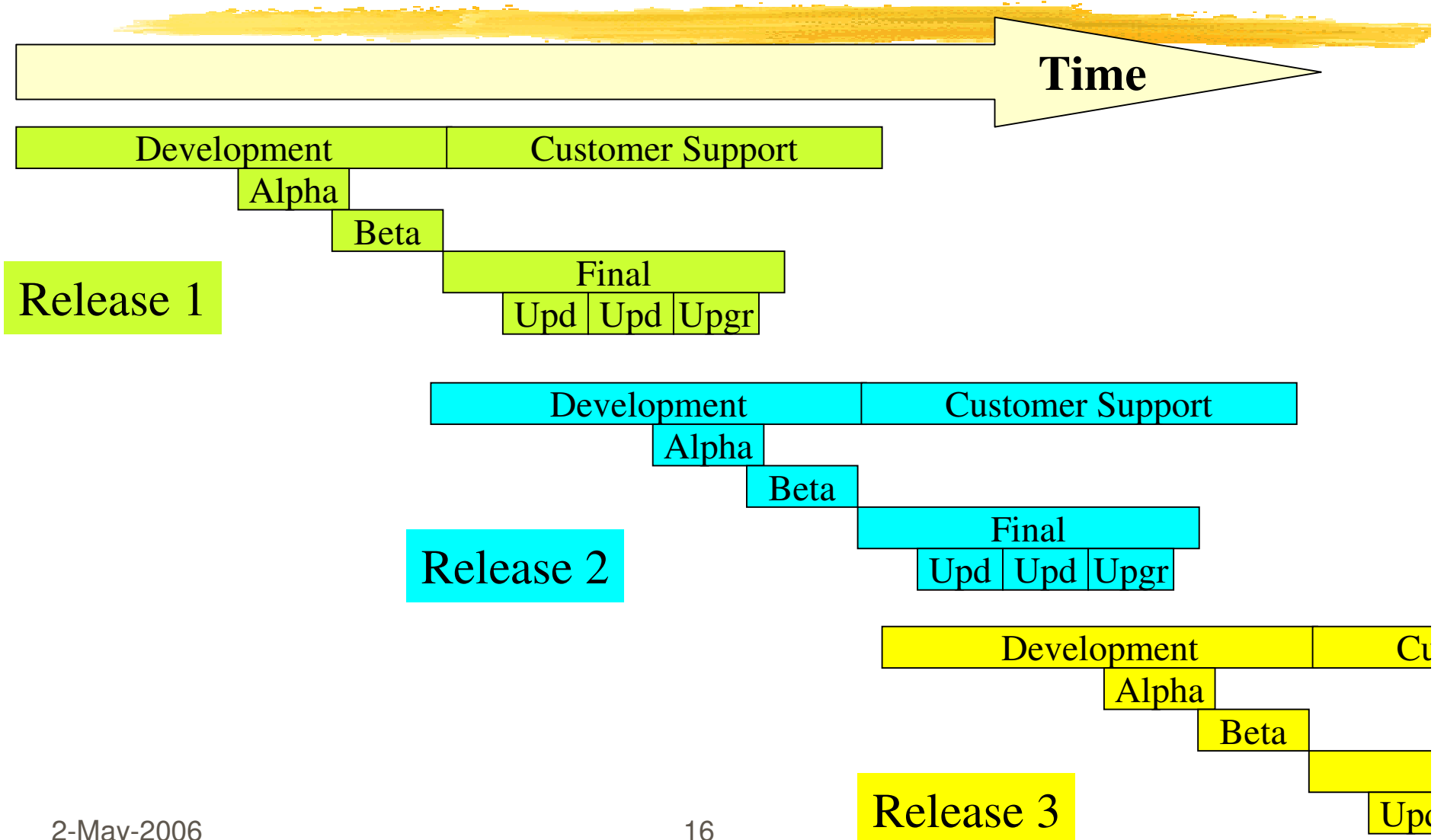


- ◆ X.Y.Z[A|Bn]
 - ◆ X: Major Release Number
 - ◆ Y: Feature Release Number
 - ◆ Z: Defect Repair Number
 - ◆ A|B: Alpha / Beta Release with a number 'n'
- ◆ Full Releases
 - ◆ Complete installation
- ◆ Partial Releases
 - ◆ Only certain components, files, executables, etc.
 - ◆ Requires a previous Full Release

Full and Partial Releases



Development & Releases



Internal Release



- ◆ Deliveries to internal users
 - ◆ Other developers (daily build)
 - ◆ Component Producer (team delivery)
 - ◆ Outsourced / Offshored Software Component
- ◆ Upcoming (external) Release Number is unclear
(Patch? Update? Upgrade? Final?)

Internal vs. External Release



- ◆ Predict Next Release (after R1.5.2)
 - ◆ Internal name is 1.5.3 (lowest possible)
 - ◆ Expected next upgrade release 1.6.0
- ◆ Special Encoding of Internal Releases
 - ◆ 1.5.3.001; 1.5.3.002 (extension)
 - ◆ Asterix.5.3 (project name encoding)
 - ◆ tj21042004.0929; dj31082005.1501

Release Distribution



- ◆ Packaging of Software Product
- ◆ Medium for distribution
 - ◆ CD; DVD; Internet; etc.
- ◆ Installation of Software
 - ◆ External installation
 - ◆ Preserve Customizations and Configurations
 - ◆ Preserve "old" storage formats (or upgrade it)
 - ◆ Internal installation
 - ◆ Different "targets" (test, debug)
 - ◆ Different Machines (development, complete product)

Homework



- ◆ Read Chapter:
 - ◆ Defect Management (needed for invited speaker)
- ◆ Read Papers:
 - ◆ ArchEvol: Versioning Architectural-Implementation Relationships
 - ◆ On Product Versioning for Hypertexts
- ◆ Prepare Lecture: Group PR7 and PR8
- ◆ Next week: UCM: Unified Change Management invited speaker Frank Schophuizen (Topic)

Conclusions



- ◆ Release Management
 - ◆ External Releases
 - ◆ Full Releases
 - ◆ Partial Releases (Upgrade and Patches)
 - ◆ Internal Releases
 - ◆ E.g. Component development
- ◆ Release Distribution
 - ◆ Packaging
 - ◆ Installation