

# SCM & Release Engineering: a consultant's hard-earned methodology revealed

D. Sandy Currier

# Outline

- Background
- The StreamTree Graph
  - ◆ Examples
- The Power of Perforce
- The Release Engineering Workflow Graph
  - ◆ Examples
- ReleasePro™
- Usage Examples

- # Background
- *The Software Development Manufacturing Problem:*
  - ◆ What is a goal of a software development enterprise?
    - ◆ *To manufacture software*
    - ◆ *With quality*
    - ◆ *With minimum expense*

# Software Manufacturing

- Two primary domains:
  - ◆ SCM – Software Configuration Management
    - ✦ Mega-deliverable: **effective Parallel Software Development Environment (PSDE's)**
  - ◆ RE – release engineering
    - ✦ Mega-deliverable: **effective automated, characterized, audited releases**

# Release engineering

- Definition
  - ◆ the engineering steps necessary to move computer files (source, derived, 3<sup>rd</sup> party) from their originating source locations to the end user

# State of the Art

- SCM is becoming well understood
  - ◆ Sophisticated tools
  - ◆ Accurate white papers
  - ◆ Reference/study materials
- Release engineering not so
  - ◆ Terminology?
  - ◆ Tools?
  - ◆ References?

# Mega-trends Driving RE Workflows

- Iterative Development Process results in more releases
- Component-based development adds component release schedules
- Maturation of SCM tools reveals RE needs

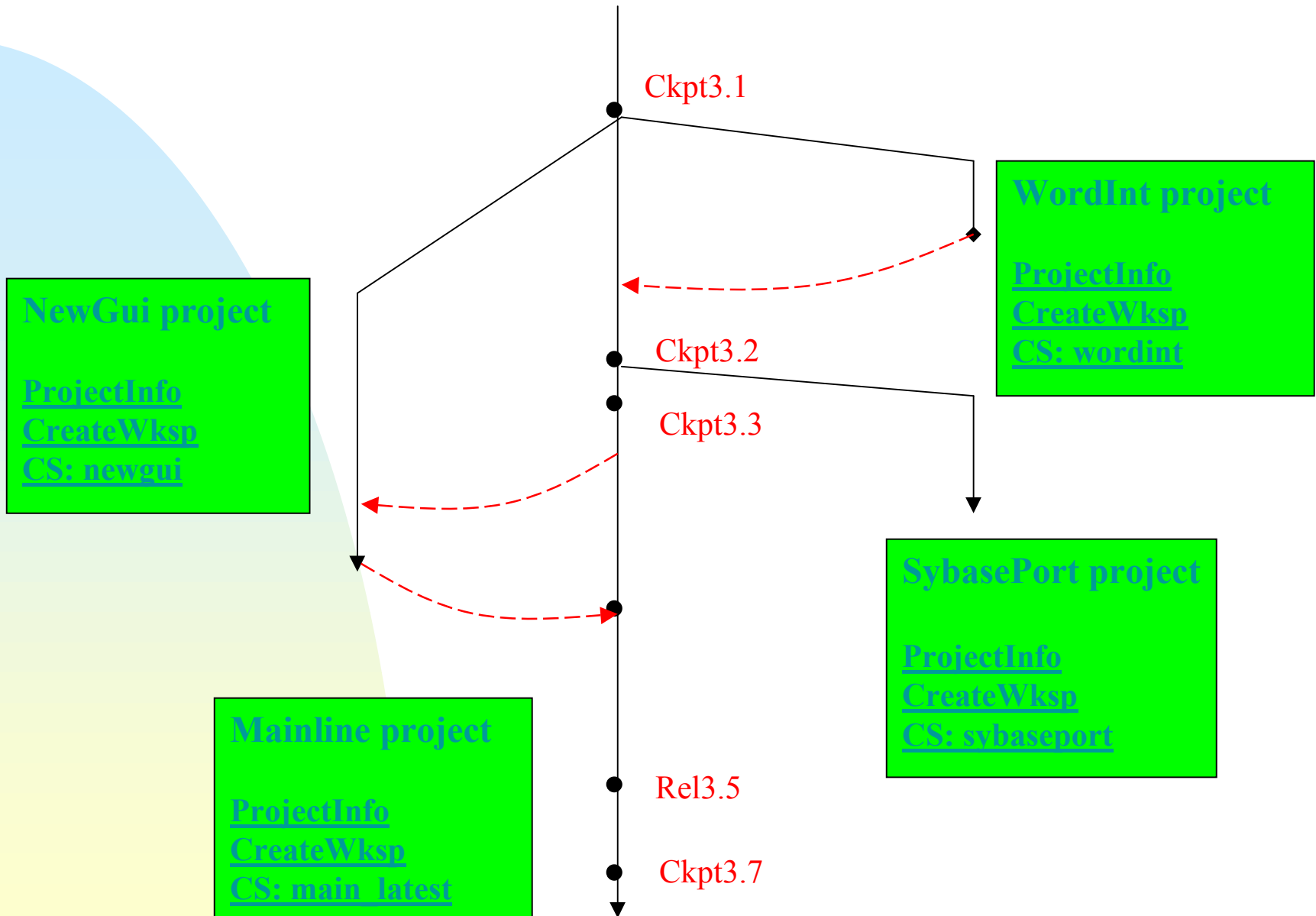
# Hard-earned Lesson

- There is a solution for each domain
- SCM
  - ◆ StreamTree graph
  - ◆ SCM tool
  - ◆ PSDE's
- Release engineering
  - ◆ RE workflow graph
  - ◆ ReleasePro™

- # The StreamTree Graph

  - 2-D graph of the codelines
  - Y-axis is code change (delta)
  - X-axis is parallelization factor
  - Each codeline a vertical solid line
  - Each branch a straight but angled solid line
  - Each merge a curved dashed line

Figure 1



# StreamTree Graph (cont.)

- More than a version tree of a single file - it is the 'sum' (union) of all the version trees of all files
- Union of all changes on all codelines
- Primarily a communication & enabling tool
- Records merging
- Contains hyperlinks

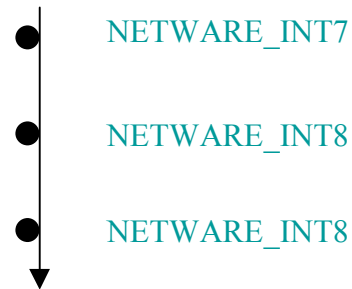
# The Power of a ST Graph



- Quickly and easily communicates the SCM 'meta' process to all.
  - ◆ Can facilitate differentiating between meta-processes and codeline specific processes.
- Describes the specific codeline process
- Helps the players to determine where changes should go
- References/coordinates future plans

# Embryonic Development

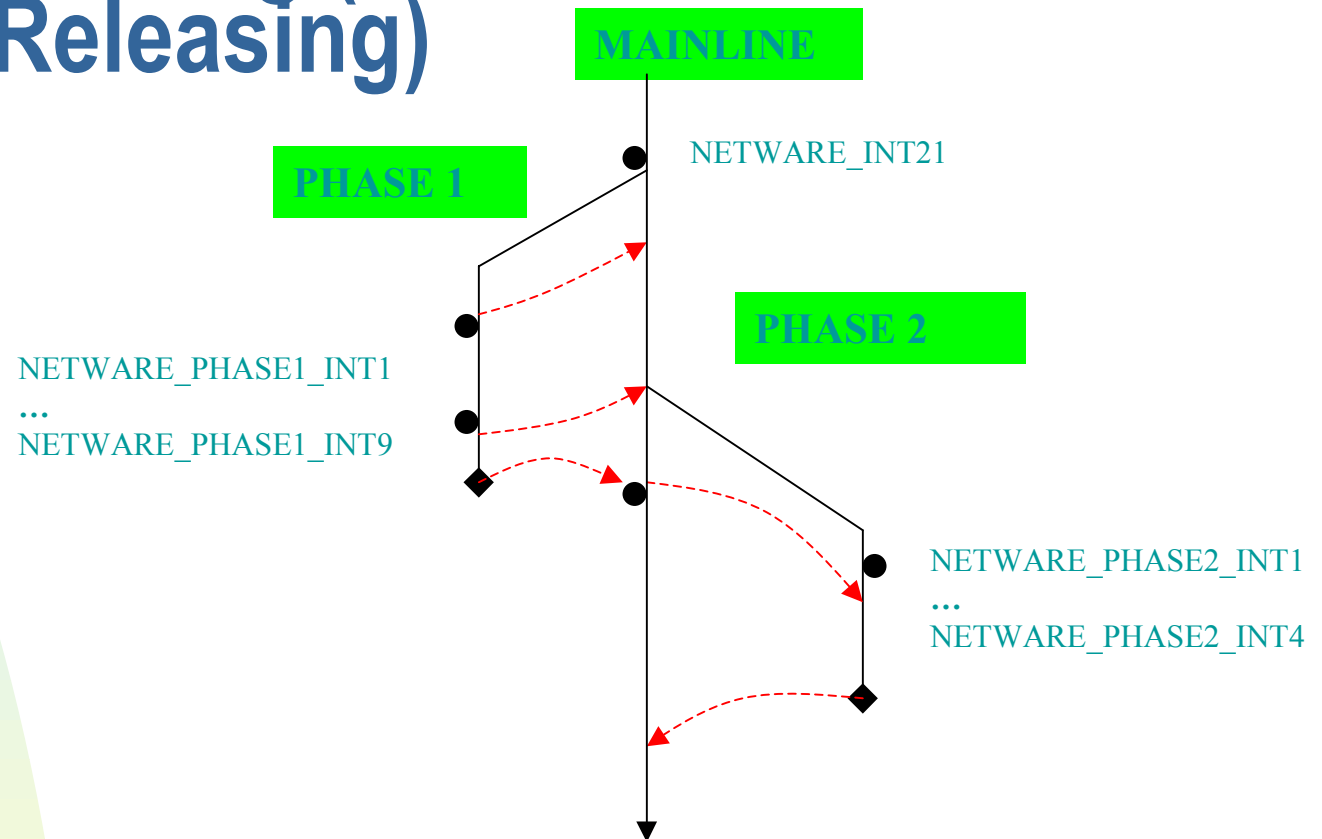
MAINLINE



## *Rules of the Road*

- All development occurs the on the mainline
- Branches are not normally employed
- Releases (to SQA or wherever) are via regular checkpoints.

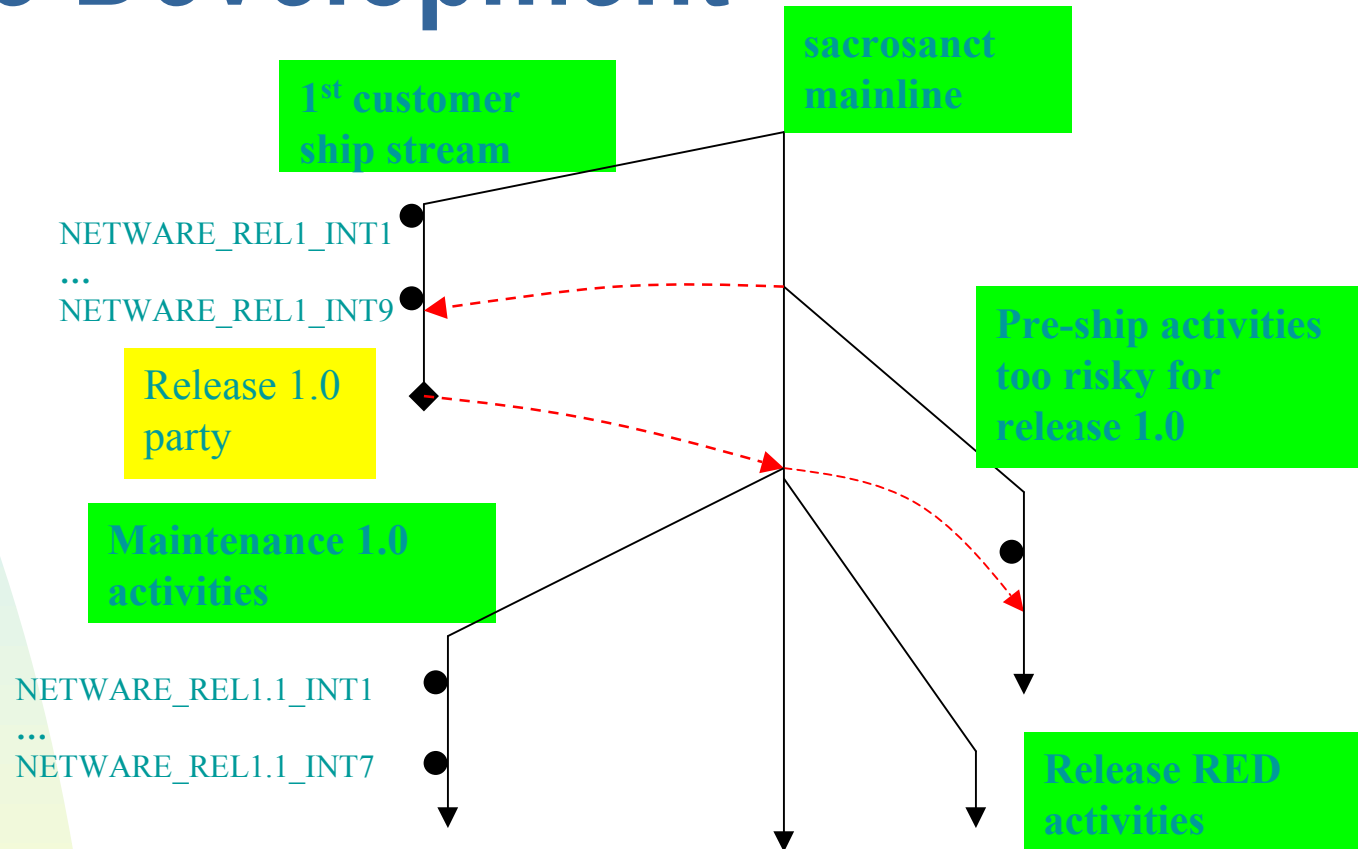
# Phased Releasing (a.k.a. Stabilized Releasing)



## Rules of the Road

- Each stabilization codeline (phase/release) is branched from main.
- Any completed and tested work on a branch is merged to the mainline.
- When a phase/release is completed, it is shutdown except for emergency/patch work.

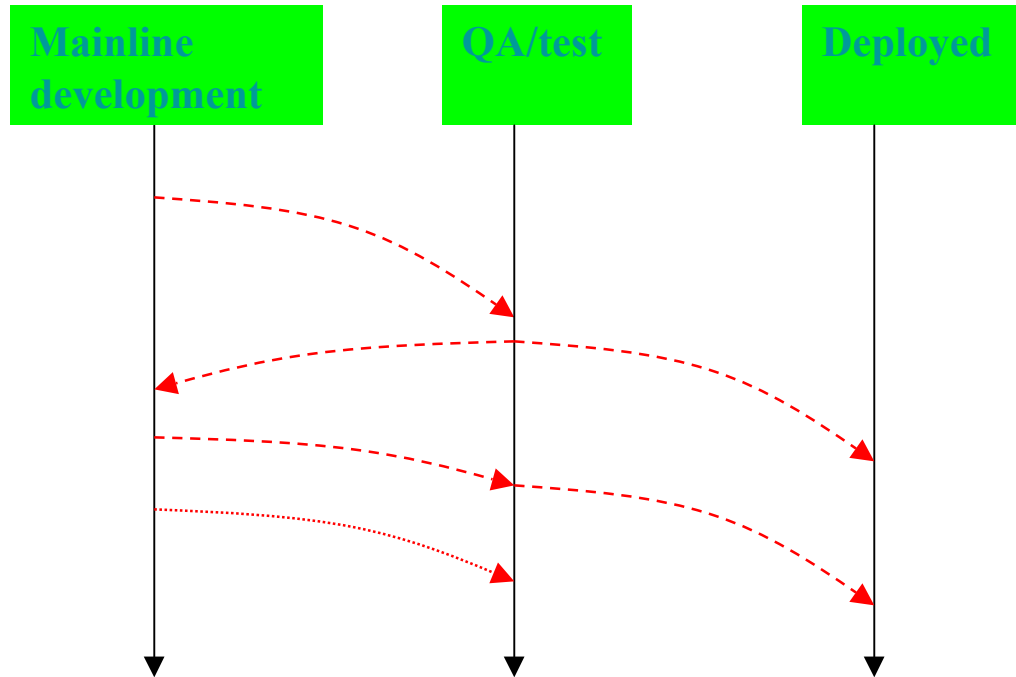
# Enterprise Development



## Rules of the Road

- The mainline is always sacrosanct: no open development, no kitchen sink.
- Release streams **MUST** rebase from the head of the mainline before releasing and then merge to after releasing.
- All streams **MUST** rebase (merge out again) with mainline as soon as possible when the mainline changes.

# Continuous Development (web-based)



## Rules of the Road

- All development occurs on the Mainline.
- Note the absence of labels or checkpoints – the Deployed codeline may be in a state of continuous update.
- There may be different *promotion* processes that are a function of directory, file type, etc. By *promoted* it is meant merging the change to the next most stable codeline.
- The default method of promotion can be a copy-merge (-at).

- # The Power of Perforce

  - Changesets/changelists
    - ◆ Associate the edits of multiple files with a single purpose
    - ◆ Reduce the amount of information
    - ◆ Can use change numbers instead of labels

# Perforce (cont.)

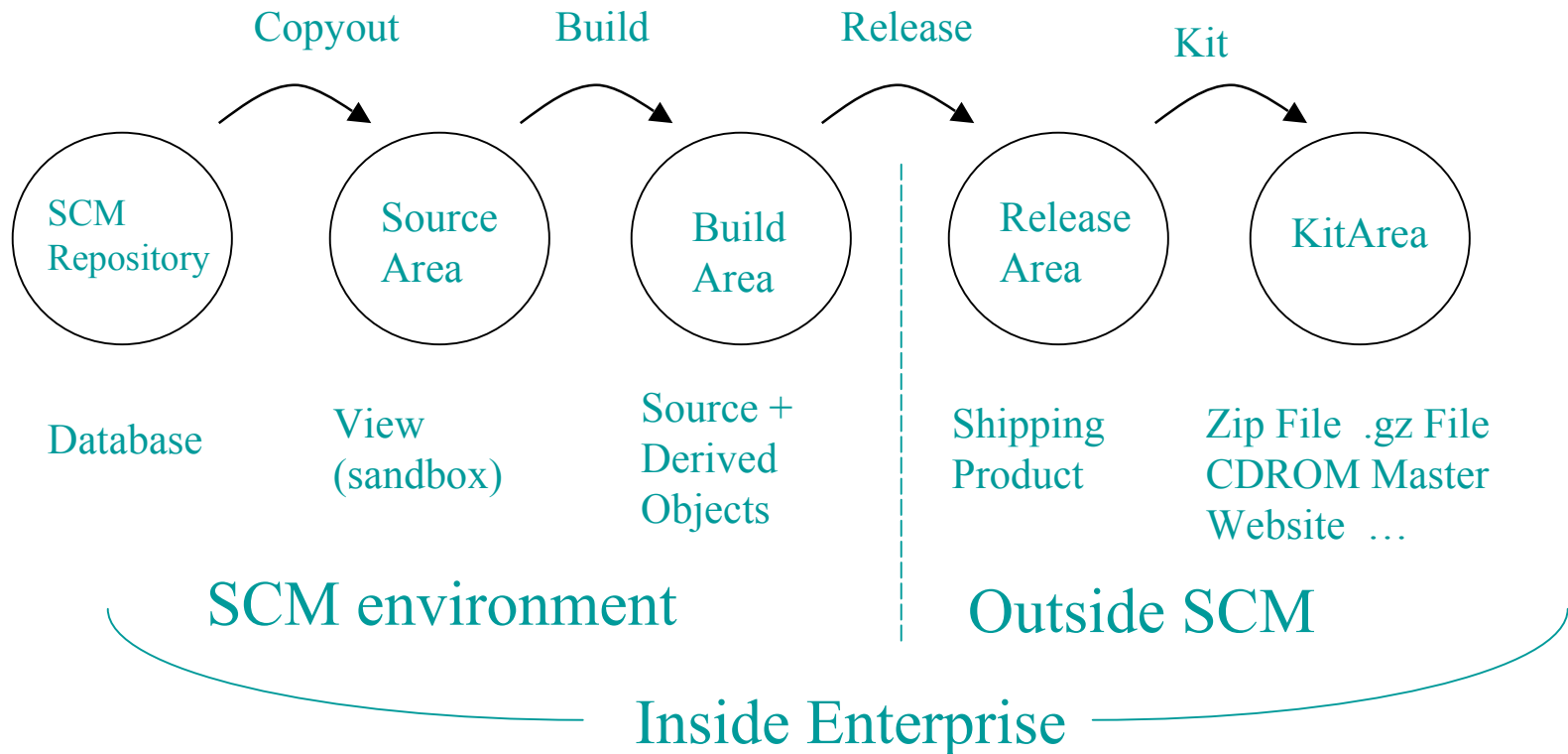
- Selective merge tracking
- Easy support of copy, ignore, and 'merge' merge
- Job query support (defect-tracking integration)
- 24x7 building (review daemon based)

- # The RE Workflow Graph

  - The release engineering (RE) workflow graph
    - ◆ Communicates information mainly to the relevant and key players
    - ◆ Describes the flow of a *dot* from the StreamTree Graph to the end user

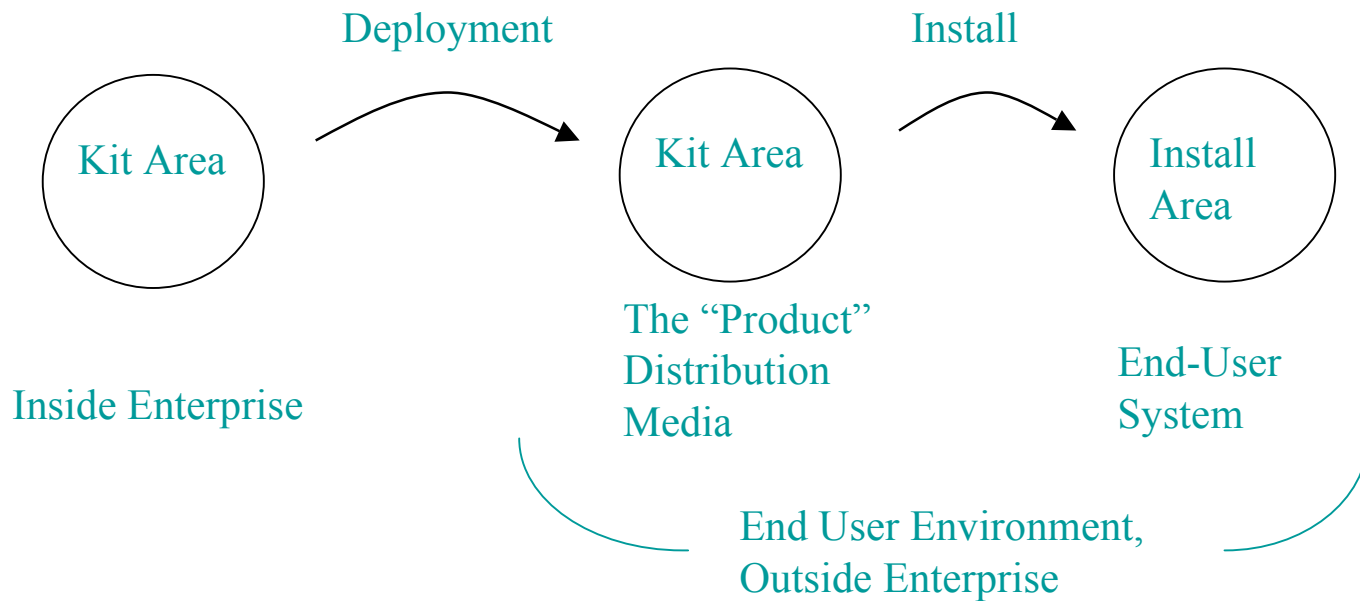
# 'Typical' Product Development (PD) Workflow

## ■ Software Production

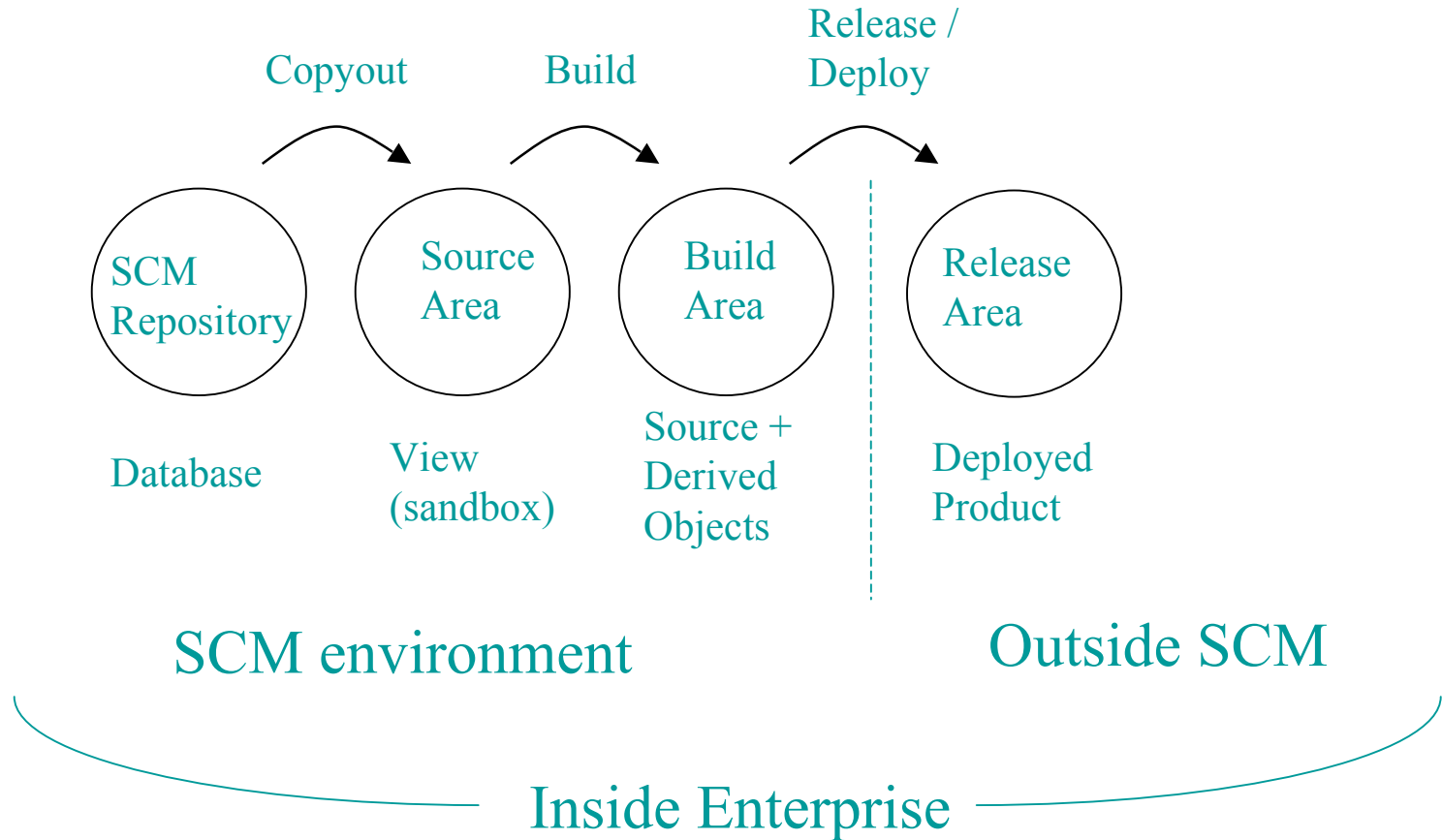


# 'Typical' PD Workflow (cont.)

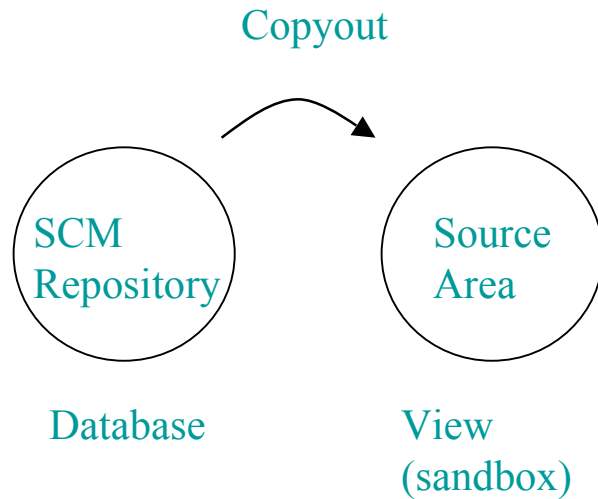
- Software Installation



# 'Typical' Information Technology (IT) Flow



# 'Typical' e-Commerce (Web) Flow



Inside SCM environment

## Caveats:

- Possibly 100% within SCM
- Possibly no build
- Promotion and SCM usage models important

# RE Workflow Notes

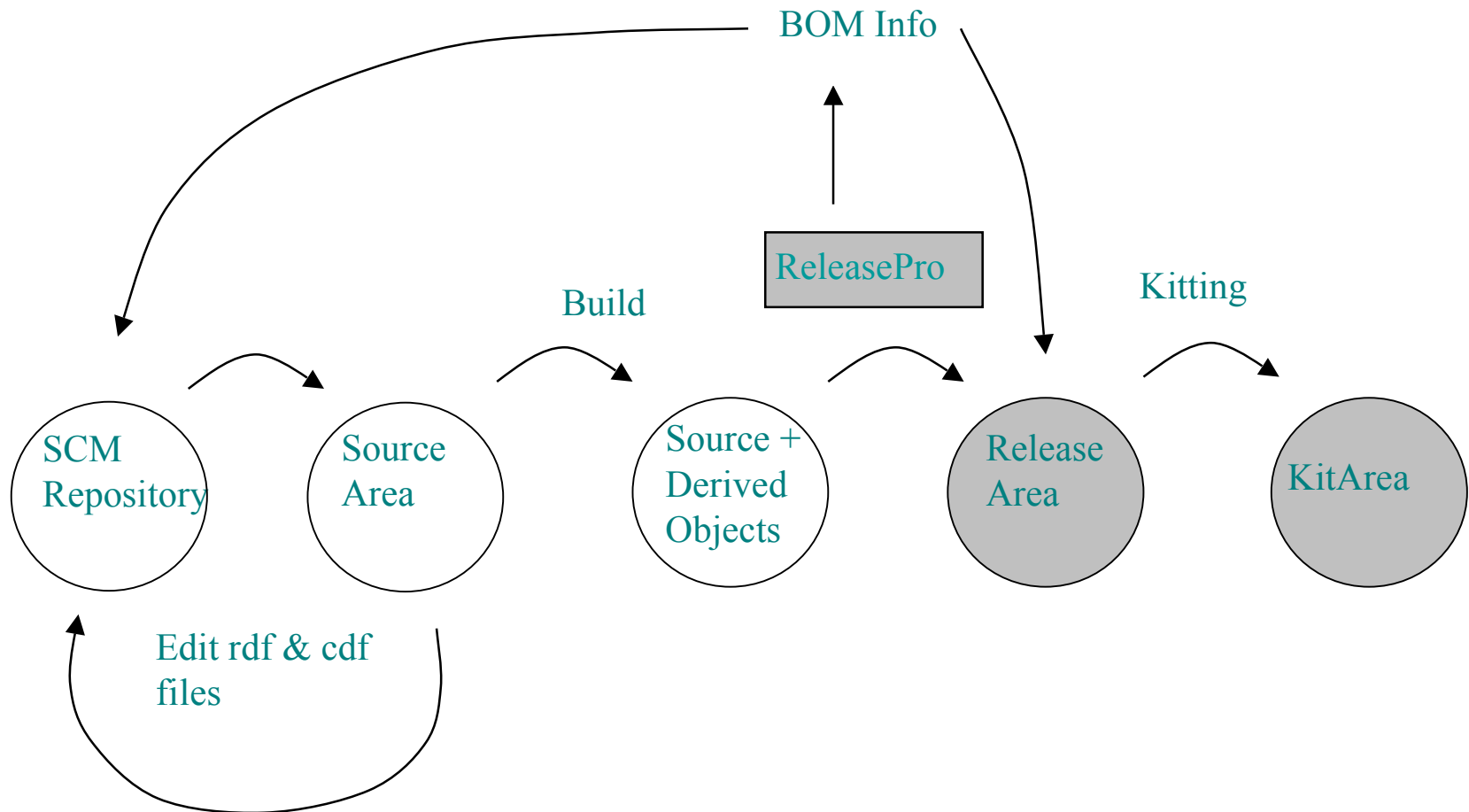
- Mapping Build Areas to Release Areas to Kit Areas to InstallAreas
  - ◆ BuildArea is optimized/organized for software development
  - ◆ ReleaseArea is optimized/organized for release-time considerations
  - ◆ InstallArea is optimized/organized for run-time considerations. May be identical to the ReleaseArea



- **ReleasePro™**

- A Release Engineering tool that automates the release phase of the RE Workflow Graph
- Powerful BuildArea to ReleaseArea file mapping (conditionalization, wildcards, etc.)
- Automated, fully characterized releasing with logging
- Simplified and repeatable release process

# ReleasePro RE Graph



# ReleasePro Features

- Integrated with Perforce, ClearCase, InstallShield, ClearQuest, ...
- Allows anyone to create a release
- Releasing occurs at machine speeds (not human)
- Creates groundwork for a sane RE Workflow support

# ReleasePro Features (cont.)

- Release lists are stored in SCM (rdf and cdf files)
- Existing SCM environment fully leveraged (tools and processes)
- Provides Bill-of-Material files and validation utilities
- When integrated with defect-tracking (ClearQuest), provides a first class release object

- **Examples using ST and RE Graphs**
- What does it mean to promote a codeline?
- What does it mean to make a release?
- Extreme Parallel Software Development: which codeline is next?
- Others examples covered in white paper

- **What does it mean to *promote* a codeline?**
  - Create a new codeline or change the policy on an existing one?
    - ◆ Use the StreamTree graph to communicate either (usually create a new one)
  - Can use the StreamTree graph (hyperlinks) to support creating Perforce clients for the new codeline.

- **What does it mean to make a release?**
  - Create a *dot* on the StreamTree graph and *push* the RE workflow button.
  - Does not effect the SCM process design or vice versa.
  - Record the release in the release management database (ReleasePro + ClearQuest).

- **Extreme Parallel Software Development: which codeline is next?**
  - Single sacrosanct codeline
  - Various development codelines all racing to release next. First one there wins.
  - StreamTree graph very important on communicating past, present, and future SCM events.

# Summary

- Hard-earned lessons:
  - ◆ solve PSDE type problems with the StreamTree Graph employing a dozen or so rules of thumb. Perforce works well.
  - ◆ automate the RE workflow with a tool like ReleasePro.
- This has allowed the author to walk away from many a successfully completed contract.

# Contact Information:

<http://www.releng.com/>

Info@releng.com

265 Davis Road

Release Engineering Inc.

Bedford, MA 01730-1515