Implementation (Development)

Manage the Chaos in Development

- Case Fred

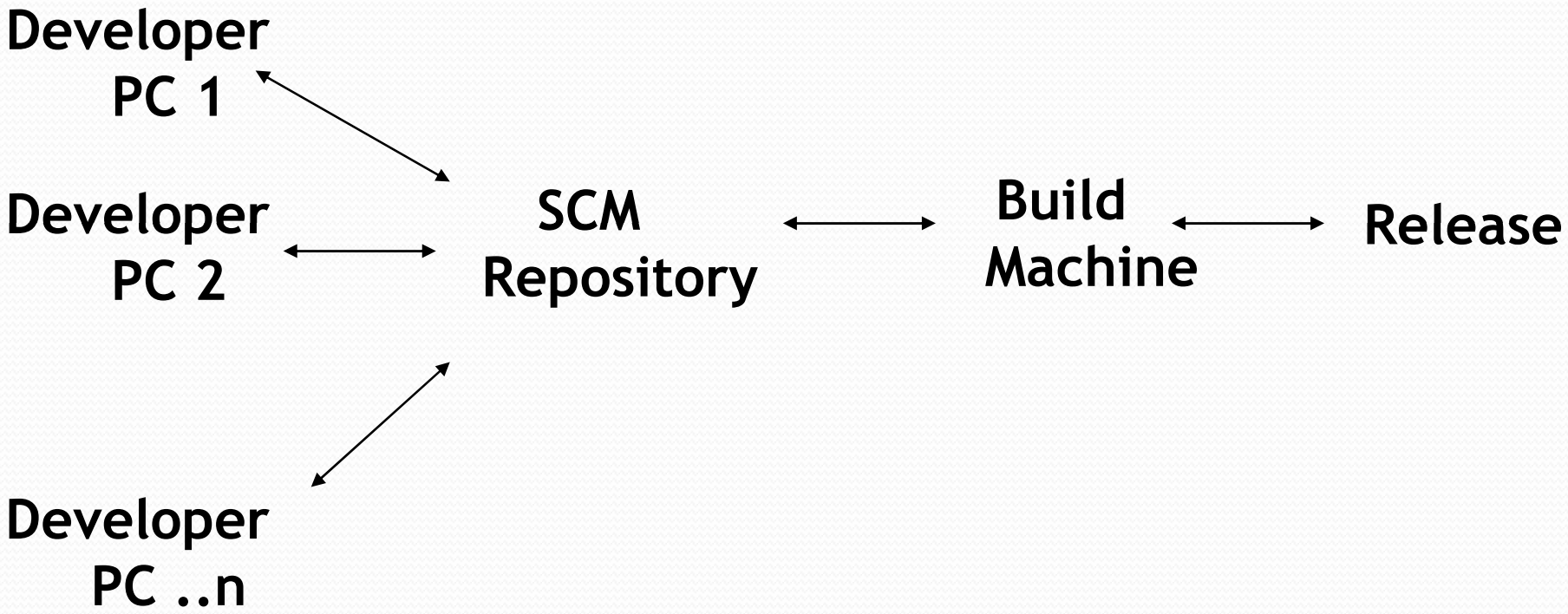
- Are these problems familiar? What can we do about them?

- Are our issues different? Which are those?
- What can we do about them?

Chaos symptoms

- Works on my machine, fails in integration!
- New bug fixed, old bug reintroduced!
- Customer is using an older version, cant locate the source!
- Accidental overwrite of latest changes, lost two days' work!
- Bugs found in integration take longer to localize and fix!

Reduce the Chaos in Development



Reduce the Chaos in Development

Version Control

Build Scripts

Track CR

Automated tests

Track Issues

Continuous Integration

Reduce the Chaos in Development

- Version Control – Use a Source Code Control System (VSS ; ClearCase, SubVersion etc.)
- Build Scripts – no step of the build process should be manual
- Continuous Integration – automatic build and test every time code changes
- Track Issues – Which customer, which version, how to reproduce, severity, priority; fix; by whom; which version
- Track Change Requests – submitted by, priority, impact on time, cost, approval status

Improve Collaboration and Learning

- When everything is changing, it is necessary to keep everybody on the same page!
- Try various ideas to improve collaboration & reduce confusion!
- SPOC (single point of contact) at customer and our ends;
- Daily Meetings – keep them short, be specific, list problems, but don't solve them (solve offline; don't reinvent wheels)
- Code change notifications – email, purpose, reviewer name
- Code Review: frequent - small amounts of code - mentor new folks

How to Prioritize – use 80-20 Pareto

- 80% of the Value comes from 20% of the Features
- Priority One: Required; you cant ship without these features
- Priority Two: Very Imp; you probably wont ship w/o these
- Priority Three: Nice to have; these wont delay the ship date
- Priority Four: Polish; add a finished feel to the product
- Priority Five: Fluff; only if you are ahead on time and budget

To summarize “Manage Deliverables”!

- Manage Requirements and find defects early
- Reduce Chaos in Development and prevent wastage
- Start with a working system; tackle high-risk areas first; don’t defer integration till the end;
- Keep filling in the pieces and keep getting user feedback early
- Improve Collaboration and share learning

Project Closure

Post Mortem Analysis

- As the name suggests, it is about Analyzing the project after it is over
- Gather all the learning's, from mistakes and from new best practices
- It is time to reflect and learn from the experience

Post Mortem Analysis

- Be specific: Discuss the details of the project, including phases, budget estimates, resource allocation, employee time, and so on.

- Document what went right and what went wrong. Were the project outcomes better than expected? What problems were encountered? Did the project take longer or less time than expected?

- Experimentation: How many risks did you take on this project? Did you try out new technology, business processes, or innovative ideas?

- Thank you!