Deployment Strategies

CS-370: Software Design and Development

Robert Bruce

What is software deployment?

- Software deployment is the act of distributing a finished software product to the customer.

Source: page 359, *Beginning Software Engineering* (2nd edition) by Rod Stephens

Deployment strategies

- **Scope**: Consider the size of the software product and the number of users it will affect. When planning a software deployment consider its impact when distributed to a large number of users.
- **The plan**: List the steps necessary for deployment including an alternative backup strategy in the event the primary deployment strategy is not successful. Anticipate what can go wrong during deployment as well.
- **Cutover**: “The process of moving users over to the new application.”
- **Staged deployment**: Build a practice area specifically for customers to run the software product to minimize catastrophic failures.
- **Gradual cutover**: Distribute the new software product to some customers while other customers continue to using their existing older system.
- **Incremental deployment**: Release the software product gradually with some new tools or features to give customers time to adjust and become familiar with the new feature or tools.
- **Parallel testing**: Have some users simulate the tasks in their job on the new software product while other users actually fulfill their tasks in their job on the existing (non-upgraded) software. The goal is to identify any possible limitations or failures with the new software in the customer’s job pipeline.

Source: pages 362 - 365, *Beginning Software Engineering* (2nd edition) by Rod Stephens

Factors which impact a LARGE software deployment

- **Physical environment**: Infrastructure where the software will be deployed such as office furniture (desks, chairs, file cabinets, cubicles, etc.), non-computer office electronics (lighting, desk lamps, coffee machines, pencil sharpeners), and supplies (paper, pens, pencils, staplers, staples, file folders, coffee, tea, etc.).
- **Hardware**: Refers to computing hardware such as network cables (ethernet, fiber optic, etc.), printers, scanners, routers, switches, desktop computers, notebook computers, computer servers, computer screens, keyboards, mice, etc.
- **Documentation**: Software product user-manuals, help guides, illustration guides, etc. in print and online form.
- **Training**: For software which requires training due to complicated features and/or a complicated interface which is radically different from legacy software it is replacing.
- **Database**: The software may require an additional database server including additional hardware to run the database management system.
- **Other people's software**: This includes software which co-exists with your software such as the operating system running on each computer or productivity tools such as scanning software, spreadsheets, email, web browser, etc.
- **Your software**: Your application and its accompanying tools.

Source: pages 365 - 366, *Beginning Software Engineering* (2nd edition) by Rod Stephens

Software deployment mistakes

- **Assume everything will work**: If it can fail, it will. Anticipate and prepare for the worst case-scenario.
- **Have no rollback plan**: When a deployment fails, revert to the previous release the customer is currently utilizing to minimize customer down-time.
- **Allow insufficient time**: Software deployment can take a significant amount of time if all does not go as planned.
- **Don't know when to surrender**: Define conditions in which software deployment will be cancelled and rescheduled for later release.
- **Skip staging**: Staging can reveal traps and pitfalls in the customer's pipeline that you – the developer - may not have realized.
- **Install lots of updates all-at-once**: Installing lots of updates all-at-once greatly increases the likelihood of incompatibility and inoperability.
- **Use an unstable environment**: Ensure the hardware infrastructure is stable before deployment. Otherwise, your software will likely fail due to flakey hardware.
- **Set an early point of no return**: This refers to rolling back the software to a previous release due to a failed deployment.

Source: pages 367 - 368, *Beginning Software Engineering* (2nd edition) by Rod Stephens