Software Project Management

Engineering best practices

BITS Pilani

Nandagopal Govindan

Agenda

- Continuous integration
- Single code base
- Product configuration capability
- API for integration
- Component based design
- Prove value, scale later
- Platform as a product

Continuous integration (CI)

- CI is a practice where developers integrate code into a shared repository frequently
- Each integration is verified by an automated build and automated tests.
- Key benefit is detecting errors quickly and locate them more easily.

Experience sharing: What benefits have you experienced using CI / CD?

Single code case

- Maintaining multiple versions of the software is challenging
- Build a core product with common features needed by target customers
- Maintain a single code base used by all customers

- Sometimes branching strategy is used to cater to different customers, one branch for one customer. But all branches have a common main trunk.

Product configuration capability

Products need customization.

Provide for

- Choice of modules (SAP)
- Configurable work flows (SaleForce)
- Configurable fields (SAP)
- Configurable rules (Navitair airline reservation)
- Configurable UI (look & feel) (Yahoo! Mail)
- Choice of language
- Configurable error messages

- Anything else?

API for integration

APIs to allow external systems to inter-operate with our product

Examples

- Facebook,
- SAP,
- Open API of banks,
- Google Maps,
- Git

Any other example?

Component based design

- Well decomposed system make a system easy to understand, build and maintain
- Web services & Micro-services are examples of components
- Components promote re-use
- They help in easier scaling & fault detection

Prove value of product, Scale later

Do not design for scale from day 1, because we do not know if the product is useful enough

Example: Zendrive

- Zendrive provides insight into driving behaviour such as how does the driver apply brake – sudden or smooth, how safe does the driver turns the vehicle – does he slow down enough before turning, etc.
- It captures data from driver's mobile phone, sends it to the server for analysis.
- Once the product value was proven, they started optimizing the product by aggregating and summarizing data on the phone and sending only summarized data to server, to reduce data transfer time and reduce processing load on central server.

Any other example?

Be open to re-architect

As business grows & expectations change, it may be necessary to re-architect the product

Example

- Amazon: Monolithic software to Micro services based software (2-pizza teams)
- Adobe Creative Suite: Desktop to Cloud
- Oracle apps: On-premise to Cloud
- Any other?

Platform as a product

- Where appropriate, build a platform
- Platform provides a base for building new services (eg Maruti Alto platform)

Example:

- Apple, Android, Firefox browser,
- AWS, Azure (databases, messaging, serverless, monitoring, etc.)
- Eclipse,
- MakeMyTrip,
- Uber,
- AirBnB (During Covid they offered adventure experiences in virtual mode)

Case study: Visio graphics-charting software (sloanreview.mit.edu)

- Scores of plug-in modules are available for Visio software that contain all types of industry- biotechnology, petroleum engineering, insurance accident reporting, and process reengineering
- Domain experts can add-in shapes (Smart shapes) and programs (charting scripts)
- These shapes carry a certain “intelligence” - automatically adjust connections between different shapes when they are moved or resized
- Major components of the platform are:
  - Core graphics engine
  - SmartShape management subsystem for incorporating and then manipulating graphic objects
  - An API that provides a standard scripting language so developers can create and integrate their own plug-in programs into Visio

Appendix