

18-642: Lifecycle & Configuration Management

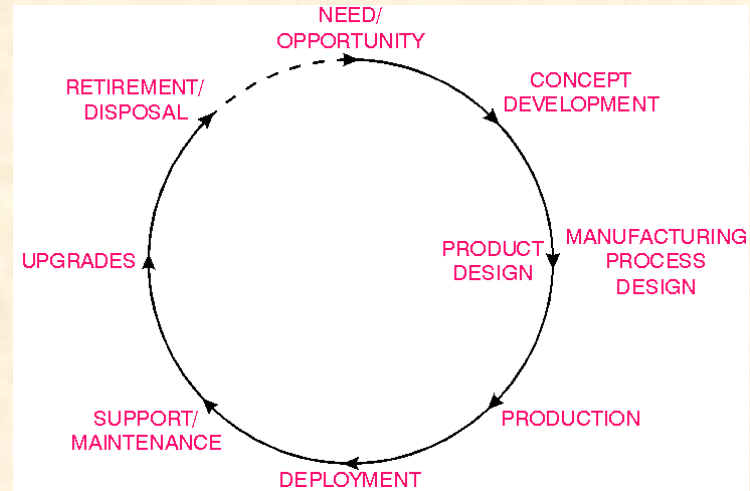
10/11/2017



Lifecycle Issues

■ Anti-Patterns:

- **No version control**
- **No configuration management**
- **Incremental features without baseline**



■ Lifecycle issues

- **Version control:** keeping different versions straight
- **Configuration management:** what's in the deployed system
- **Lifecycle:** old embedded systems (almost) never die
 - Spare parts for obsolete systems
 - Mid-life upgrades

Airbus confirms software configuration error caused plane crash

Airbus A400M flight recorder data confirms "quality issue" in setup caused failure.

SEAN GALLAGHER - 6/1/2015, 12:38 PM



A prototype of the Airbus A400M at the 2010 Farnborough Airshow.

As **Ars reported on May 19**, Airbus had issued a warning to its military customers about a potential software problem in the engine control software for the A400M. The release of the exact cause of the crash, however, had been delayed because a Spanish magistrate placed the flight data recorders from the aircraft under seal. Airbus has since been able to obtain the flight data, which Lahoud said confirms that the engine control software had been improperly configured during the installation of the engines on the ill-fated aircraft.



https://www.youtube.com/watch?v=TUiX6m6WLDY&ab_channel=Sciences

<http://arstechnica.com/information-technology/2015/06/airbus-confirms-software-configuration-error-caused-plane-crash/>

Version Control

■ Stores snapshots of versions

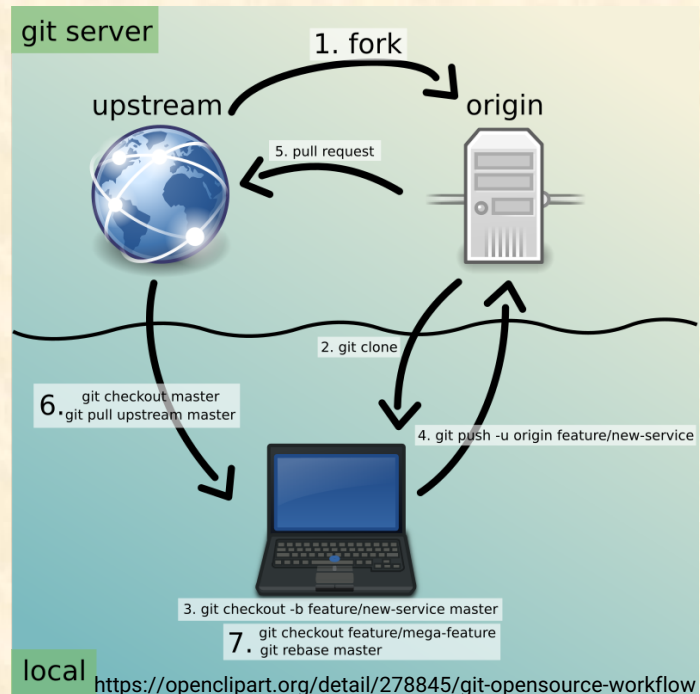
- Ability to roll-back to previous version
- Synchronizes compatible versions
 - Development vs. release versions
 - “Branch” and “merge” parallel efforts

■ Many popular tools – use one!

- Watch out for binary file management
- Multi-site can require special care

■ Beyond the obvious...

- Also version: design documents, requirements, tests, tool chain
- Need process for who/what/when to update version
- Needs to tie into disaster recovery (have you tested it?)



Configuration Management

■ CM is identifying a particular version

- Which version has which bug fixes?
- Which version has which features?
- Which version should be the next release?
- Which version is in a failed product?

■ Beyond the obvious...

- Make sure returned units can self-report version
 - Which library version, which driver version, etc.
 - Need SW version, HW version, config data version (if applicable)
 - Attempts to track this in a central database never work 100%
- Need to know which SW is compatible with which HW
 - Need to be able to re-create a build, which might require obsolete tools
- Feature activation: which features have been licensed by user?



Airbus A-380 bolt with part tracking information. (Size: 2 cm x 1 cm)

Gas Pump Startup Printout Example)

```
-----  
Thank You  
For Shopping At  
Wilkins  
SHOP 'n SAVE Express  
-----  
,
```

```
*****  
* LOADED *  
SUCCESSFULLY  
* PAPER *  
*****
```

CLAMSHELL PRINTER BOARD

** CONFIGURATION **

```
- Revision : - V0507  
- CHKS/CRC : (C2B1) 5517h  
- S/N : yy/ww-ssss  
- Pre Heating : on  
- Paper Temp. : high  
- Auto Advance : on  
- Watchdog : on  
- Opto Jam s/h: CC/99  
- Opto Pass s/h: 28/0A
```

** SERIAL INTERFACE **

```
- Parameters : n,8,1  
- Flow Control : Dtr/Dsr  
- Baud Rate : 38400  
- Level I/O : TTL
```

```
!"#$%&'()*+,-./01234567  
89:;<=>?@ABCDEFGHIJKLMNO  
PQRSTUVWXYZ[\]^_`abcdefg  
hijklmnopqrstuvwxyz{|}~
```

Embedded Systems Live (almost) Forever

■ 10-50 year lifecycles are common:

- Example: SAGE air defense system
 - Started 1954; deployed 1963
 - Vacuum tubes; 500,000 lines of code
 - In operation until 1983
 - 1AESS telephone switch:
1976 through 2008+ (maybe some still working)
 - Cars routinely run for 15-20 years



■ Challenges

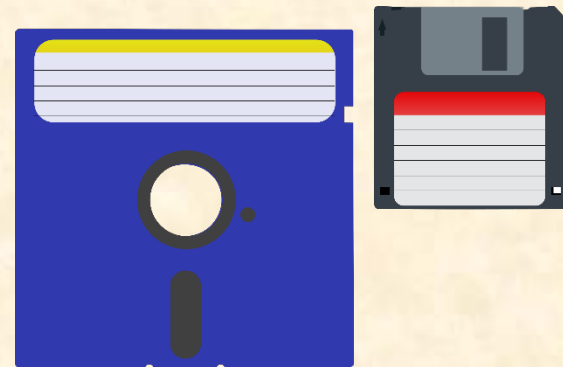
- End-of-life for hardware & software
- Adding new services to existing platform
- Security problems if not updated



Best Practices for Lifecycle Management

■ Take Configuration Management seriously

- Version control is just a starting point
- Also need to track what version goes where
- Plan for 2x the lifecycle you think will happen
 - And have a plan for product end of life



■ Lifecycle pitfalls

- Releasing the wrong version
 - Development version, debug version (e.g., watchdog turned off), etc.
- Cutting corners on software configuration management
 - What if you can't reconstruct software involved in a field failure?
- Getting caught without replacement parts
 - Pay attention to end-of-life buys
 - Anticipate obsolete: hardware, media, tools, libraries, ...

