Quiz

1. Which are the five essential properties of software?
2. What are accidental properties of software?

1

The staged model of software evolution

CS 515, Spring 2020

Laura Moreno
lmorenoc@colostate.edu

2

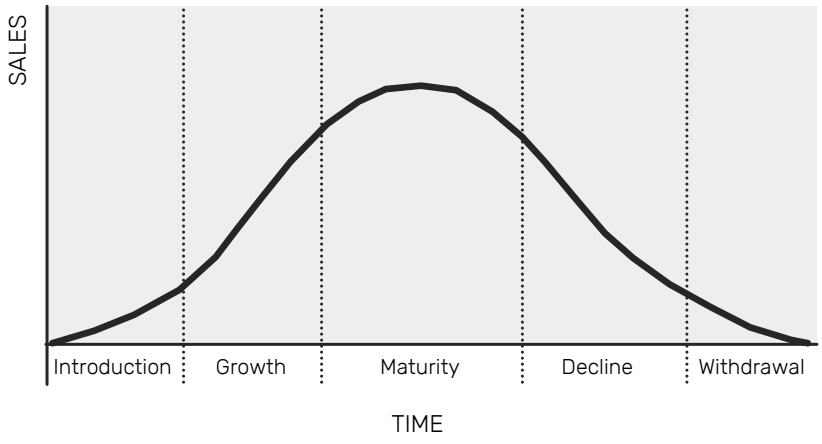
Traditional software life cycle models

Traditional software life cycle models

Product life cycle

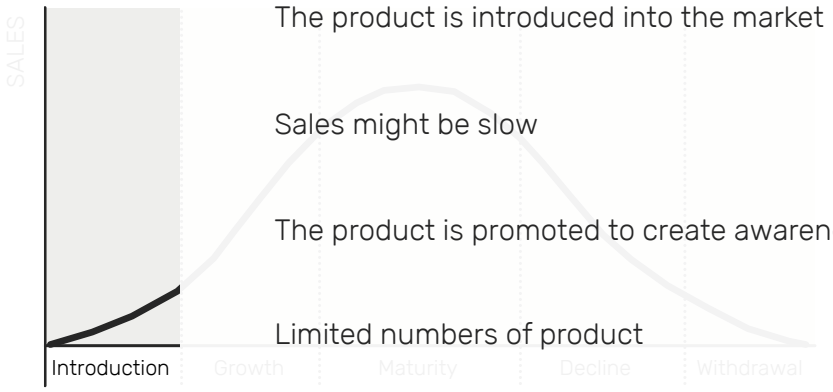
13

Product life cycle



14

Product life cycle >> Introduction



The product is introduced into the market

Sales might be slow

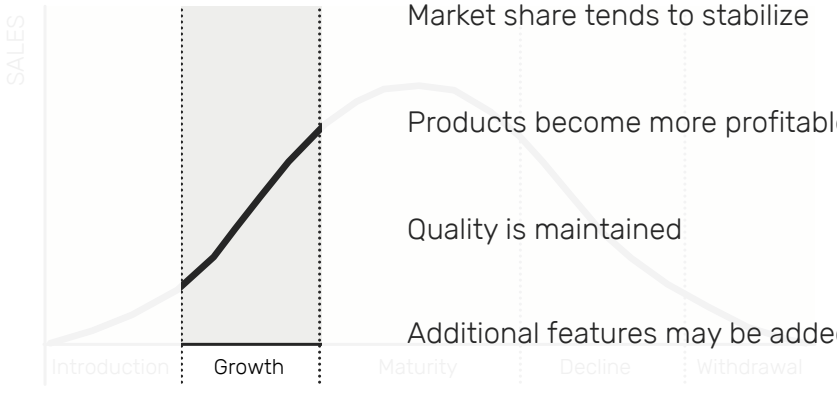
The product is promoted to create awareness

Limited numbers of product

Patents and trademarks are obtained

15

Product life cycle >> Growth



Market share tends to stabilize

Products become more profitable

Quality is maintained

Additional features may be added

Competition might appear

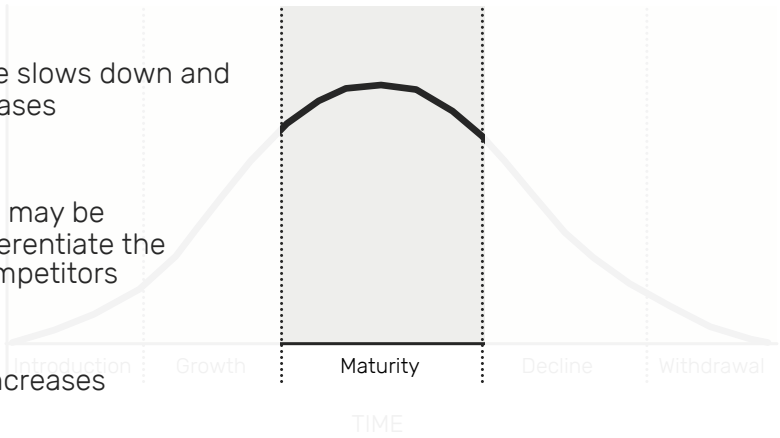
16

Product life cycle >> Maturity

Sales growth rate slows down and eventually decreases

Product features may be enhanced to differentiate the product from competitors

Advertisement increases



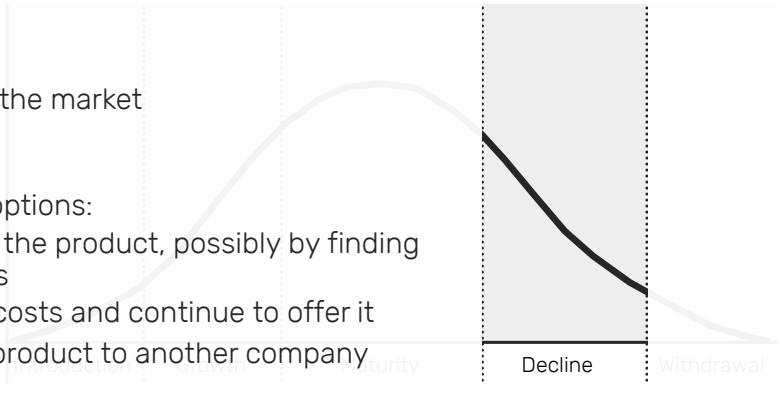
17

Product life cycle >> Decline

A downturn in the market

Management options:

- Maintain the product, possibly by finding new uses
- Reduce costs and continue to offer it
- Sell the product to another company

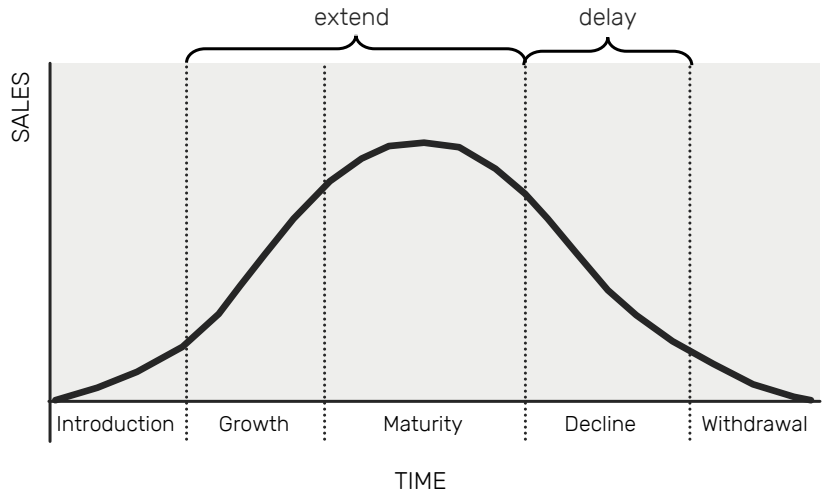


18

Product life cycle >> Withdrawal

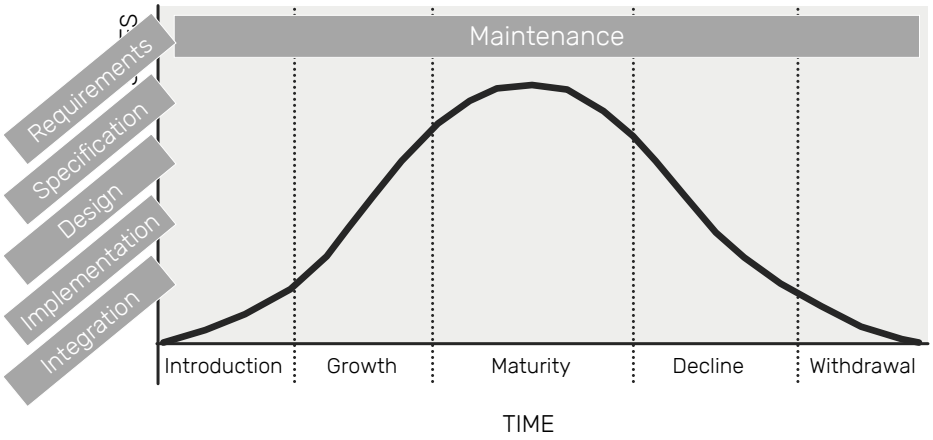
19

Sale strategies



20

Product life cycle vs. Traditional software models

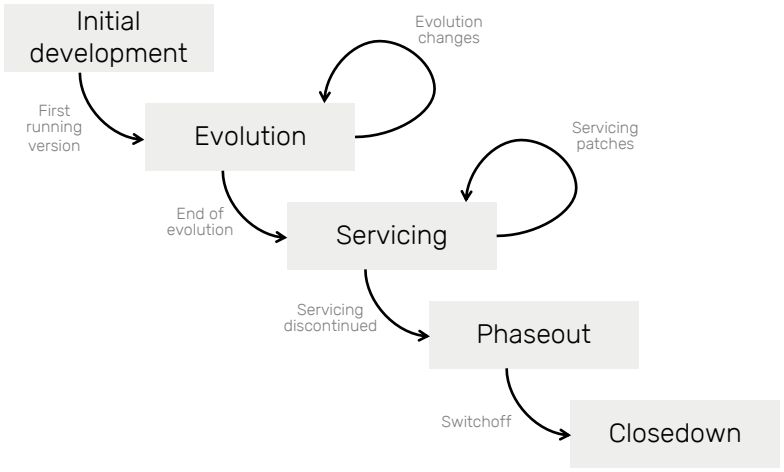


21

Staged model for software life cycle

22

Simple staged model



23

Stage 1 >> Initial Development

Initial development

Develop the first functioning version

Establish software team expertise

Establish system architecture

Heavy investment

Impending deadline

24

Stage 2 >> Evolution

Most often the software is released during evolution

Adapt/correct the capabilities and functionality of the system based on experience with software

Architecture slowly loses its original lucidity and integrity

Market demand is strong, and revenue is good

25

Stage 3 >> Servicing

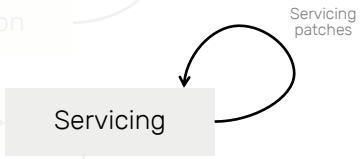
Coherent architecture or/and expertise are lost

Architecture limits the scope of possible changes

Minor repairs and simple functional changes

Revenue begins to fall

Whether to improve the code depends on cost/profit ratio



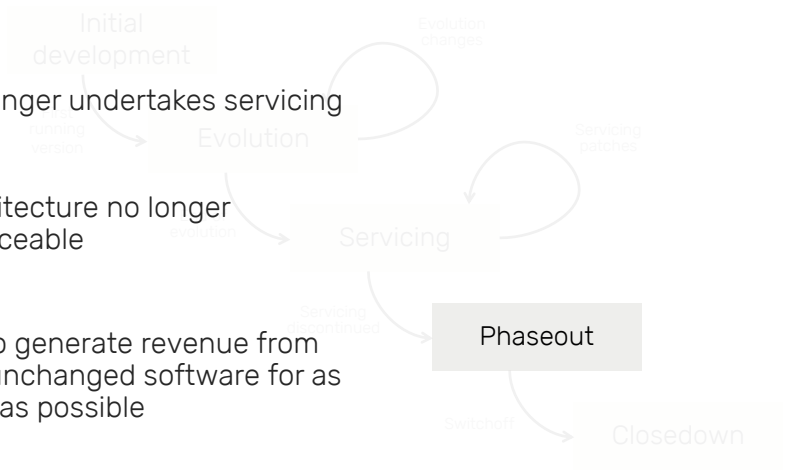
26

Stage 4 >> Phaseout

No longer undertakes servicing

Architecture no longer serviceable

Try to generate revenue from the unchanged software for as long as possible

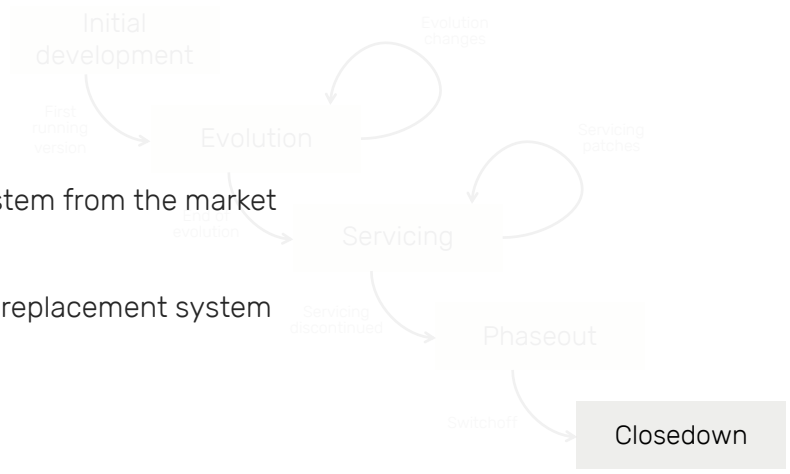


27

Stage 5 >> Closedown

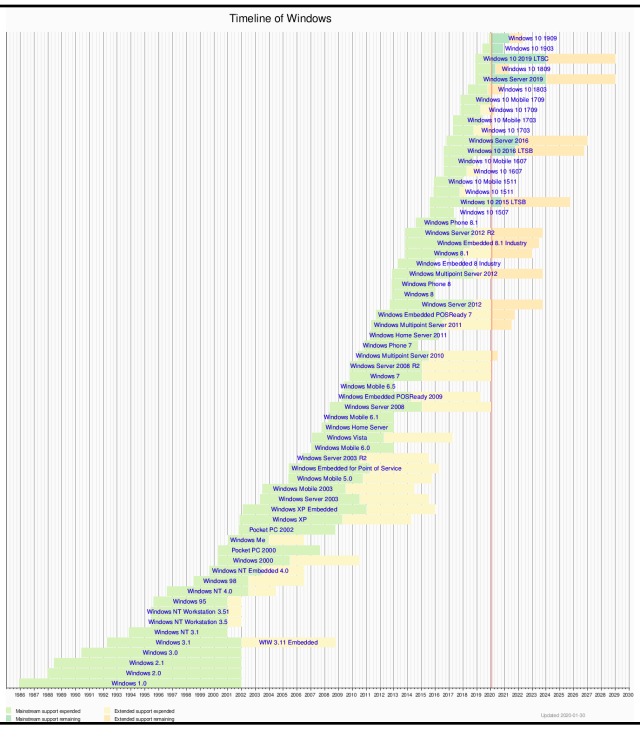
Withdraw the system from the market

Direct users to a replacement system



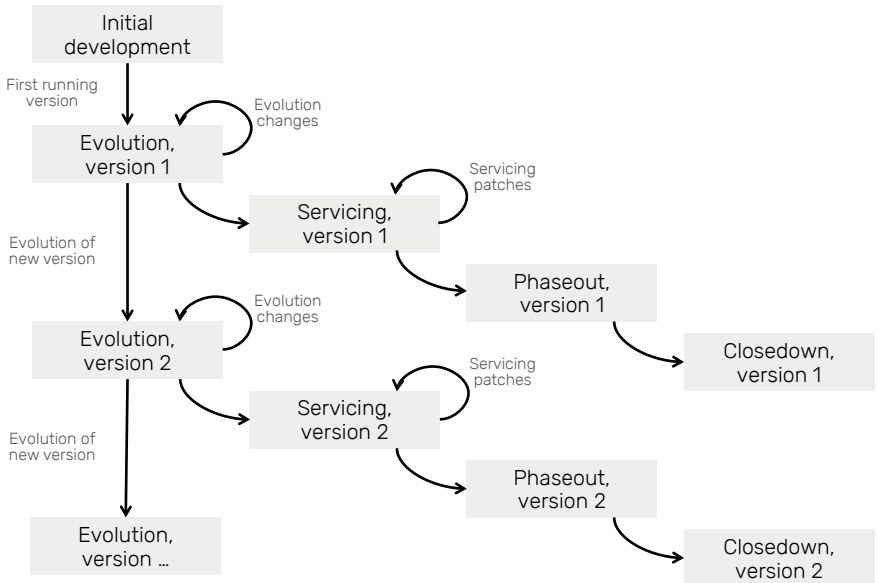
28

An example: MS Windows



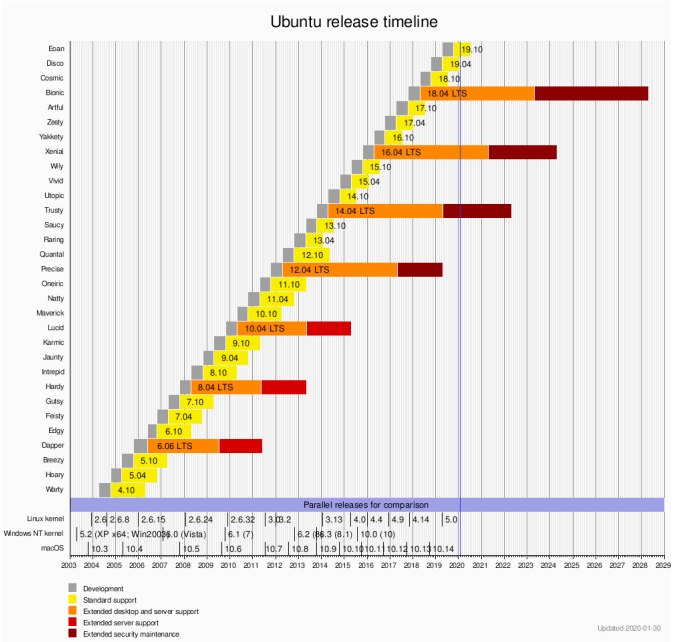
29

Versioned staged model



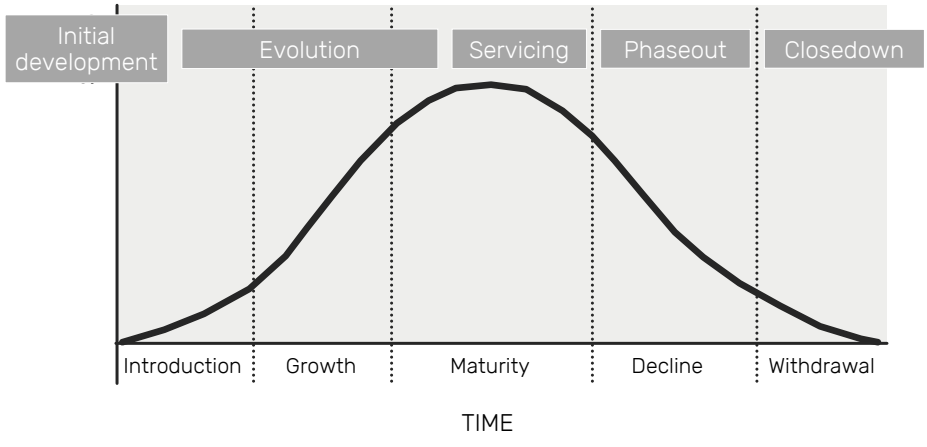
30

Another example: Ubuntu



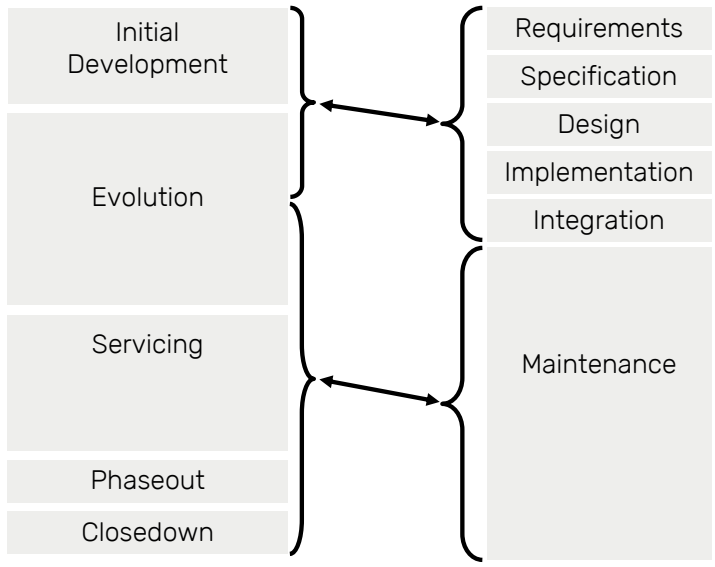
31

Staged model vs. Product life cycle



35

Staged model vs. Traditional models



Importance of (better) software models

Each stage has different needs

- Technical
- Procedural
- Staff
- Management

Once we understand the nature and needs of each stage, we can:

- Identify best solutions for each stage
- Explore the boundaries of each stage
- Analyze the information flow, which will enable managers to plan better
- Explore options to keep a system in a particular stage for as long as possible

Reading

Rajlich, V, Bennett, K, A Staged Model for the Software Life Cycle, IEEE Computer, July 2000, pp 66-71