

03: The context of software development

Software Project Management
Philippe Kruchten

Jan. 2014 Copyright © 2005-14 by KESL 1

Outline

- Two models:
 - What is common
 - What is different
- One size does not fit all
- The vast spectrum of software development projects “cases”
- Sources of Variability
- The Octopus

Jan. 2014 Copyright © 2005-14 by KESL 4

Conceptual model

- Unifying model
- Commonality
- Variability

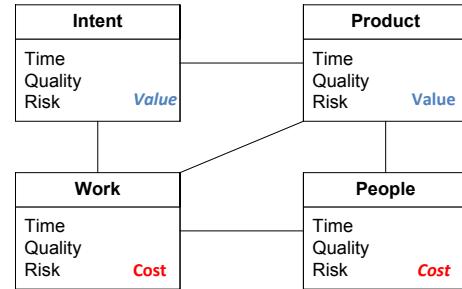


Usage:

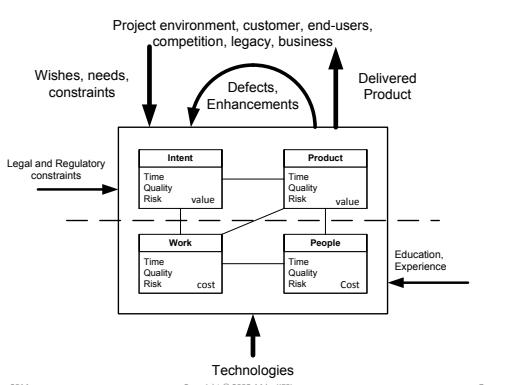
- Set up the syllabus
- Analyze and critique practices
- Compare approaches, process models

Jan. 2014 Copyright © 2005-14 by KESL 5

Frog: “All projects are the same”



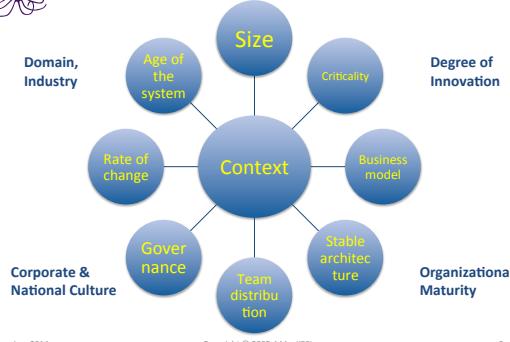
Jan. 2014 Copyright © 2005-14 by KESL 6



Wishes, needs, constraints
Legal and Regulatory constraints
Project environment, customer, end-users, competition, legacy, business
Defects, Enhancements
Delivered Product
Education, Experience
Technologies

Jan. 2014 Copyright © 2005-14 by KESL 7

Octopus: “All projects are different!”



Domain, Industry
Age of the system
Rate of change
Governance
Corporate & National Culture
Size
Criticality
Business model
Stable architecture
Team distribution
Organizational Maturity
Degree of Innovation

Jan. 2014 Copyright © 2005-14 by KESL 8

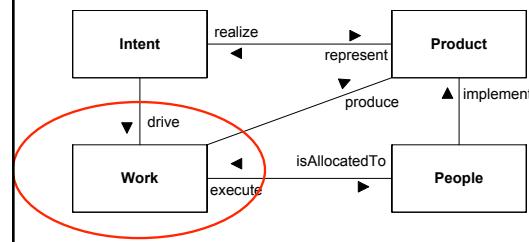
Defining Process

- Process: set of activities intended to achieve a goal
- Process to run a project
 - = software development process
 - = software engineering process
 - = software process
- Software project management is a subset of that software development process

Copyright © 2005-14 by KESL

9

Process is defining Work



Jan. 2014

Copyright © 2005-14 by KESL

10

Importance of process

- Processes represent the collective knowledge on how to run a project
- A Project process has 2 main aspects:
 - Engineering : design test, code, technology
 - Project management
- Good project managers must understand the process
 - Both the engineering process
 - and the management process

Jan. 2014

Copyright © 2005-14 by KESL

11

Type of Software Projects

- Commercial, speculative vs. contract work
- Many instances vs. single few instances
- Internal vs. external (acquisition)
- Greenfield vs. evolution or legacy transformation
- Single project vs. program or portfolio
- Size
- Process: one size does *not* fit all...!

Jan. 2014

Copyright © 2005-14 by KESL

12

Project Context

- Environmental Conditions (organization)
 - Drive/constrain
- Context Attributes (software project)
 - Selection and adaptation
- Good Practices (actual process)

Jan. 2014

Copyright © 2005-14 by KESL

13

Environmental conditions

- Business domain
 - E-commerce
 - Manufacturing
 - Automotive
 - Aerospace
- Number of instances
 - One, A dozen, Millions, SaaS,...
- Maturity of organization
 - Small start up
 - Mid size software Dev. Co.
 - Large system integrator
 - +... collective experience



Jan. 2014

Copyright © 2005-14 by KESL

14

Environmental conditions (cont.)

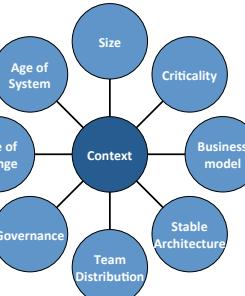
- Level of innovation
 - New product, never been done... or
 - Old classic, just better, faster, larger, ...
- Culture
 - Communication
 - Trust
 - Shared mental models
 - Education (?)

In general, environmental conditions are proper to the organization, and common to several projects



Context attributes

1. Size
2. Criticality
3. Age of system
4. Rate of change
5. Business model
6. Stable architecture
7. Team distribution
8. Governance



Jan. 2014

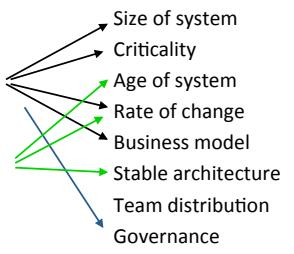
Copyright © 2005-14 by KESL

15

16

Copyright © 2005-14 by KESL

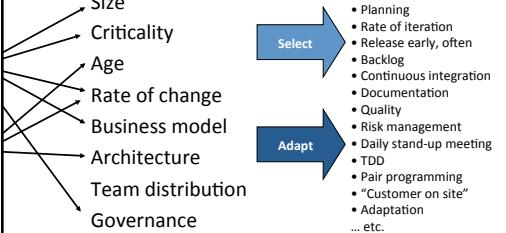
Environmental drivers → Context attributes



Copyright © 2005-14 by KESL

17

Context attributes → Practices/process



Good Practices:

- Planning
- Rate of iteration
- Release early, often
- Backlog
- Continuous integration
- Documentation
- Quality
- Risk management
- Daily stand-up meeting
- TDD
- Pair programming
- “Customer on site”
- Adaptation
- ... etc.

Copyright © 2005-14 by KESL

18

1. Size of system

- SLOC, FP
- Impacts team size, duration...

Driven by: Business domain

Related to: Legacy, geographic distribution, governance

Affects: Iteration rate, planning, communication modalities, documentation, risk management, “customer on site”, etc.



Jan. 2014

Copyright © 2005-14 by KESL

19

2. Criticality

- “Software that kills”
 - + Massive losses, damage to environment
- Demonstrably correct
- Formal methods
- Extensive testing
- Audited by external agencies

Driven by: Business domain

Related to: Rate of change

Affects: Documentation, testing, inspection



Belleville-sur-Loire, France

Jan. 2014

Copyright © 2005-14 by KESL

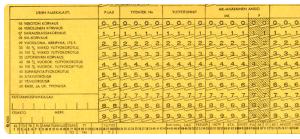
20

03 - Context

3

3. Age of system

Legacy evolution
Brownfield vs. green field development, maintenance



- *Related to:* size
- *Affects:* Testing, (lack of) documentation, architecture

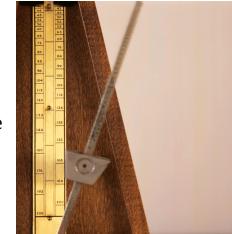
Jan. 2014

Copyright © 2005-14 by KESL

21

4. Rate of Change

- Adaptation versus anticipation
- External & internal changes
 - Customer, competitor, technology, legislators, inside organization, turnover, team evolution, maturation, ...
- *Driven by:* business domain
- *Related to:* size
- *Affects:* Iteration length, planning, adaptation,...



Jan. 2014

Copyright © 2005-14 by KESL

22

5. Business Model

- Commercial market
- Bespoke software
- In-house development
- Open source development
- ... etc.
- *Driven by:* Business domain
- *Related to:* Governance
- *Affects:* Documentation, number of instances, "customer on site", communication, risk management, geographic distribution, *rate of iteration*



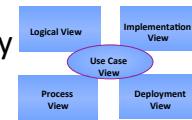
Jan. 2014

Copyright © 2005-14 by KESL

23

6. Architecture stability

- How much of a stable system and software architecture is in place when the project starts?
- *Driven by:* Level of innovation
- *Related to:* Size of system, age of system
- *Affects:* Rate of iteration, risk management, testing



Jan. 2014

Copyright © 2005-14 by KESL

24

7. Geographic Distribution of the Team

- Seems to make everything a bit harder and more susceptible to fail
- *Driven by:* Maturity of organization, culture
- *Related to:* Size, business model
- *Affects:* Communication modalities, documentation, DSM, governance, ...



Jan. 2014

Copyright © 2005-14 by KESL

25

8. Governance

- **Structural:** chains of authority, responsibility and communication to empower the various actors
- **Dynamic:** measures, control, mechanisms, policies to enable all actors to carry out their respective responsibilities

Clay Williams, IBM, 2008



Is this "big process" coming by the back door?

Jan. 2014

Copyright © 2005-14 by KESL

26

The agile “sweet spot”



System Size	• 0 ..12 ... 300
Criticality	• Simple, \$ losses , ... deaths
System Age	• Exploratory, greenfield , legacy maintenance
Rate of change	• Low, medium, high
Business model	• In house , Open Source,
Stable architecture	• Stable , changed, new
Team distribution	• Collocated , ..., ..., offshore outsource
Governance	• Simple rules , ..., SOX, ...

Jan. 2014

Copyright © 2005-14 by KESL

27

Examples: analyzing context

Business context	Commercial, speculative	Contractual	Research, Open-source
# of instance	1	A few	many
Specifier Developer relationship	Internal	Internal offshore	External, outsourced
Reuse	Greenfield	Brownfield	Legacy
Dev. context	Single	Program	Portfolio
Size of dev. team	<12	12 - 20	>20

Jan. 2014

Copyright © 2005-14 by KESL

28

Business context	Commercial, speculative	Contractual	Research, Open-source
# of instance	1	A few	many
Specifier Developer relationship	Internal	Internal offshore	External, outsourced
Reuse	Greenfield	Brownfield	Legacy
Dev. context	Single	Program	Portfolio
Size of dev. team	<12	12 - 20	>20

Jan. 2014

Copyright © 2005-14 by KESL

29

Business context	Commercial, speculative	Contractual	Research, Open-source
# of instance	1	A few	many
Specifier Developer relationship	Internal	Internal offshore	External, outsourced
Reuse	Greenfield	Brownfield	Legacy
Dev. context	Single	Program	Portfolio
Size of dev. team	<12	12 - 20	>20

Jan. 2014

Copyright © 2005-14 by KESL

30

Business context	Commercial, speculative	Contractual	Research, Open-source
# of instance	1	A few	many
Specifier Developer relationship	Internal	Internal offshore	External, outsourced
Reuse	Greenfield	Brownfield	Legacy
Dev. context	Single	Program	Portfolio
Size of dev. team	<12	12 - 20	>20

Jan. 2014

Copyright © 2005-14 by KESL

31

Business context	Commercial, speculative	Contractual	Research, Open-source
# of instance	1	A few	many
Specifier Developer relationship	Internal	Internal offshore	External, outsourced
Reuse	Greenfield	Brownfield	Legacy
Dev. context	Single	Program	Portfolio
Size of dev. team	<12	12 - 20	>20

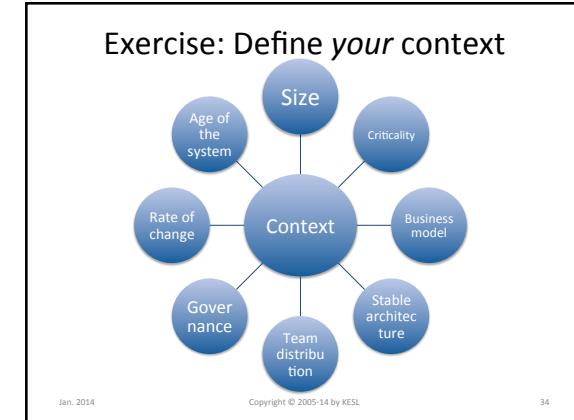
Jan. 2014

Copyright © 2005-14 by KESL

32

Business context	Commercial, speculative	Contractual	Research, Open-source
# of instance	1	A few	many
Specifier Developer relationship	Internal	Internal offshore	External, outsourced
Reuse	Greenfield	Brownfield	Legacy
Dev. context	Single	Program	Portfolio
Size of dev. team	<12	12 - 20	>20

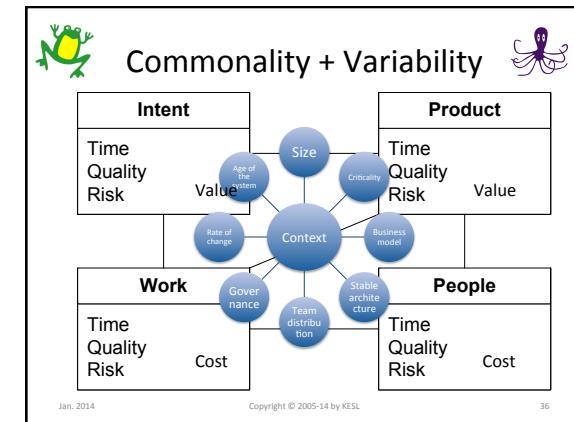
Jan. 2014 Copyright © 2005-14 by KESL 35



Summary

- Two models:
 - Commonality (frog)
 - Variability (octopus)
- Reasoning about project management
- Selection of appropriate process:
 - Lifecycle (time line)
 - Activities (work)
 - Roles (people)
 - Artifacts, workproducts (product, intent, quality)

Jan. 2014 Copyright © 2005-14 by KESL 35



A **project** is all the **work** that **people** have to accomplish over **time** to realize in a **product** some specific **intent**, at some level of **quality**, delivering **value** to the business at a given **cost**, while resolving many **uncertainties and risk**.

All aspects of software projects are affected by **context**: size, criticality, team distribution, pre-existence of an architecture, governance, business model, which will guide with practices will actually perform best, within a certain domain and culture.

Jan. 2014 Copyright © 2005-14 by KESL 37