

Requirements

Time in Phases for Different Development Models (pg. 47)

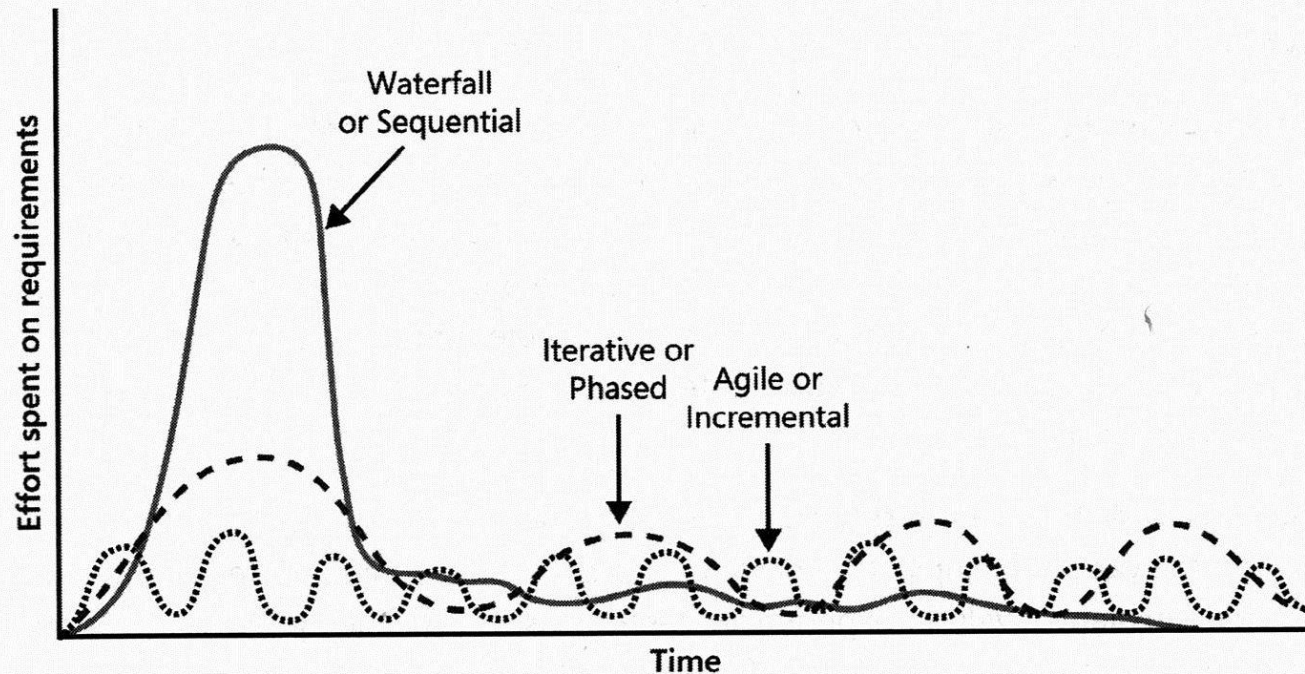


FIGURE 3-3 The distribution of requirements development effort over time varies for projects that follow different development life cycles.

Requirements

Requirements should be:

- Complete
- Consistent
- Unambiguous
- Quantitative
- Feasible to test

Informal Requirements

Informal requirements:

1. The ERROR light comes on for PSIs of -1 or less
2. The UNDERPRESSURE light comes for PSIs between 0 and 20
3. No light comes on for PSIs between 21 and 35 (normal operating conditions)
4. The OVERPRESSURE light comes on for PSIs of 36 or greater

Improvement

Informal requirements:

1. The ERROR light comes on for PSIs of -1 or less
2. If an air pressure value of -1 or lower is received by the display sensor, then the ERROR light shall be enabled, and all other lights shall be disabled.

Improvement

2. The UNDERPRESSURE light comes for PSIs between 0 and 20

If an air pressure value between 0 and 20 (inclusive) is received by the display sensor, then the UNDERPRESSURE light shall be enabled, and all other lights shall be disabled.

Shall, should and Will (page 209)

shall – requirement, desired functionality, system capability, imperative

should – desired

will – design expectation, declarative, something that is true not that developers must implement

Functional & Non-Functional Requirements

Functional – “State what a system should do.”

Non-functional requirements – “State what a system should be.” These include:

- Quality attributes
- Constraints
- External interface requirements

Acceptance Criteria

Acceptance criteria are the “conditions that a software product must satisfy to be accepted by a user, customer or other stakeholders.” (Microsoft Press)

Acceptance criteria should be developed as a joint effort between the development team and the product owner.

<https://medium.freecodecamp.org/the-acceptance-criteria-for-writing-acceptance-criteria-6eae9d497814>