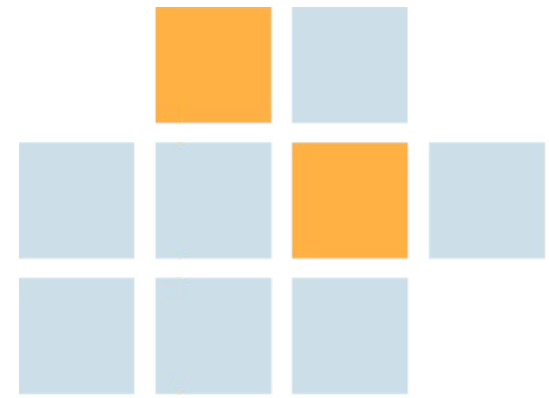


9LivesData

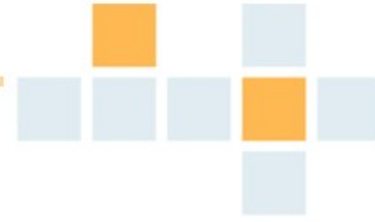
Optimizing Software Quality in B2B systems

by Maciej Przewoźnik

www.9livesdata.com

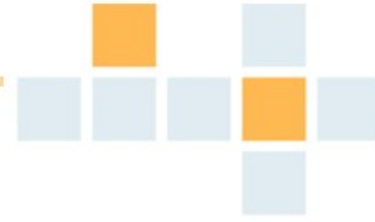


Agenda



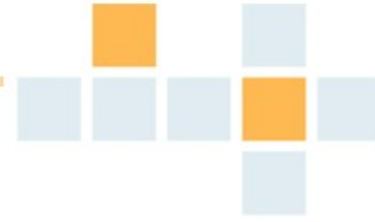
- Software quality (SQ) elements
- Software quality dependencies
- Optimizing software quality
 - Definition
 - Requirements negotiations
 - Priorities of quality attributes impact
 - Tailoring development effort
 - Test type impact
 - Development process impact
- Case study
- Summary

SQ elements > Drivers of software quality



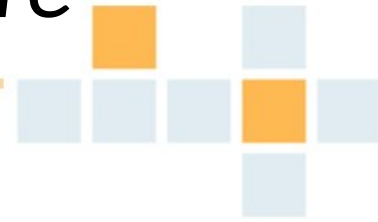
- Customer satisfaction
- Impact on user bottom line
- Company reputation
- Extensibility and maintainability of the software
 - Long-term sales
- Low quality costs avoidance

SQ elements > Costs of low quality



- Higher customer support
- Defect elimination
- Lower development speed
- Lost opportunities
- Litigations
- Bad media coverage
- Reputation loss
- Damage to customer business
- Potential people death or injury

SQ elements > High vs. low quality software



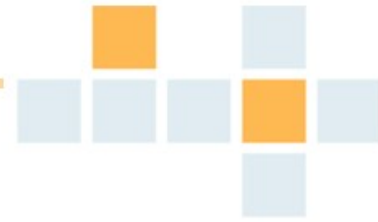
- Costs related to low quality often too high
- High quality software in fact cheaper than low quality software
- High quality to be achieved in optimal or suboptimal way
- Quality may vary for different software attributes and subattributes
 - Attributes: performance, reliability, portability, ...
 - Attributes have different priorities and requirements
 - Standardized software quality attributes: ISO-25010

SQ elements > ISO-25010 software quality attributes and subattributes



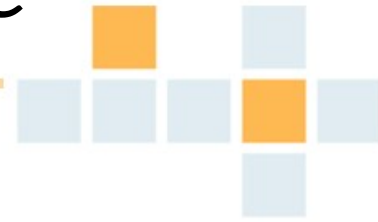
- Functional stability: completeness, correctness, appropriateness
- Performance efficiency: time behavior, resource utilization, capacity
- Compatibility: co-existence, interoperability
- Usability: appropriateness recognizability, learnability, operability, user error protection, user interface aesthetics, accessibility
- Reliability: maturity, availability, fault tolerance, recoverability
- Security: confidentiality, integrity, non-repudiation, authenticity, accountability
- Maintainability: modularity, reusability, analysability, modifiability, testability
- Portability: adaptability, installability, replaceability

SQ elements > Software quality dilemmas



- Schedule vs. features vs. quality
- What is more important
 - Customer bugs or no bugs and no customers?
 - Time to market or time to reach good quality
 - Next deadline or long term maintainability?
- Solution: aim at software quality optimization
 - For product type and stage
 - By prioritizing ISO-25010 software quality attributes and subattributes

SQ dependencies > product type and stage



Software quality depends on

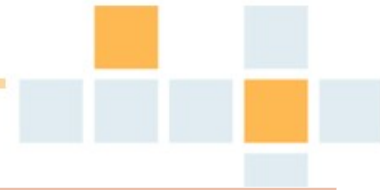
Product type

- Non-critical
- Critical
 - Safety critical
 - Business/mission critical
 - Security critical

Stage

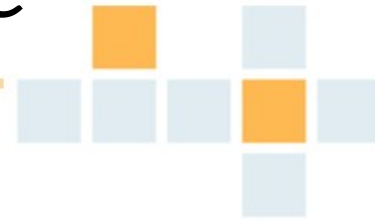
- POC (technology validation)
- MVP (business need validation)
- Beta
- Production
 - limited release (a few clients)
 - general release (many clients)

SQ dependencies > product types



| System type | Examples | Possible consequences of failure |
|---------------------------|---|--|
| Non-critical | <ul style="list-style-type: none">- company machine learning system | <ul style="list-style-type: none">- delayed business optimization |
| Safety critical | <ul style="list-style-type: none">- medical systems- flight traffic controller systems | <ul style="list-style-type: none">- damage to people health- property/resources damage or loss |
| Mission/business critical | <ul style="list-style-type: none">- online banking- stock-trading system | <ul style="list-style-type: none">- significant economic costs- loss of business- damage to reputation |
| Security critical | <ul style="list-style-type: none">- health records database- storage systems | <ul style="list-style-type: none">- sensitive data loss or theft |

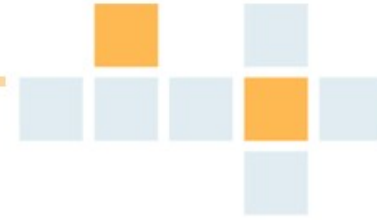
SQ dependencies > product type and stage



Quality grows with the stage

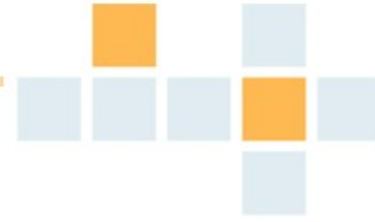
- In POC, business concept verification for investors is crucial
- In MVP, demonstrability for investors or early customers is crucial
 - Early customers expected to be forgiving and likely to give feedback
- In beta, some quality defects may be tolerable
- In production, quality is by definition high
 - Higher in general release than in limited release

SQ dependencies > common non-critical and critical software failure consequences by stage



- In POC
 - Demo delay, unproductive presentations
- In MVP
 - Demo failure, bad impression on investors or early customers
 - However, sometimes imitation by competitors or reputation loss
- In beta
 - Potential customers distrust
- In production stage (limited release)
 - Canceled contracts
- In production stage (general release)
 - Similar to limited release, but also
 - Diminished reputation (media attention)
 - Costly customer damage repair and troubleshooting

SQ dependencies > critical software failure consequences by stage



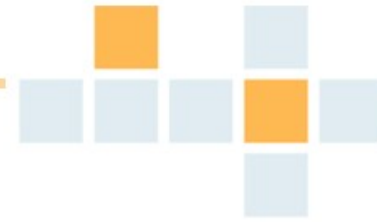
- In production stage (limited release)
 - Human life or health loss
 - Sensitive data loss or theft
 - Financial loss
 - Severe reputation loss (media attention)
 - Costly litigations
 - Running out of business
- In production stage (general release)
 - Similar to limited release, but higher risk and impact

SQ dependencies > Negotiable aspects of quality



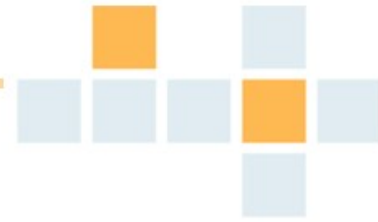
- Low-priority software attributes (ISO-25010)
- Prioritization differ by individual project
 - Priorities influenced by project type and stage
- Examples of negotiable aspects:
 - For functional stability
 - Solution precision
 - For storage: deduplication and compression ratio
 - For machine learning: accuracy
 - For reliability
 - Corner case situations
 - Lower availability (e. g. with periodic rebooting)
 - For usability
 - Substandard visual aspects

Optimizing SQ > Definition



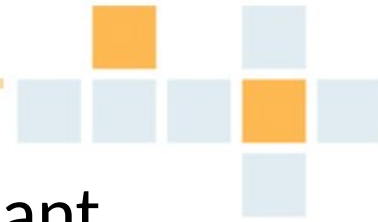
- Never about making quality low
- Requirements negotiations
- Test types, development process and development effort tailored to
 - Product type
 - Stage
 - Priorities of ISO-25010 quality attributes
- Goals
 - Balance costs and benefits for the project
 - Optimize company performance on the market

Optimizing SQ > Requirements negotiations



- Understand the motivation for the project
- Try to propose relaxed requirements
- May be the biggest cost reduction and therefore the best quality optimization

Optimizing SQ > Tailoring development effort



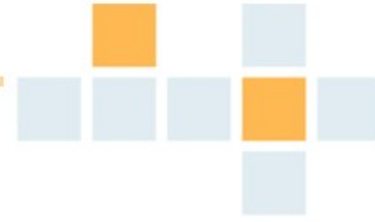
- Smaller development effort needed for less important quality attributes ISO-25010
- Examples of areas of limited development requirements for some projects:
 - Portability
 - Security
 - Internationalization

Optimizing SQ > Tailoring test type and development process



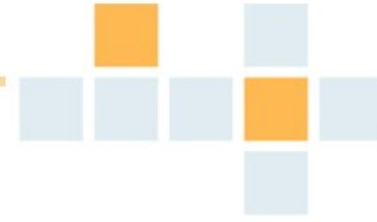
- Classification of test type and development process (general tendency)
 - Must: very high Return on Investment (ROI), limited exceptions possible
 - Should: high ROI, numerous exceptions
 - Could: generally low ROI, sometimes useful
 - Won't: activity has poor ROI
- Level: result of cost-benefit analysis
- Classification dependent on individual projects
- Classification of test type and development process influenced by priorities of project quality attributes

Optimizing SQ > Test types



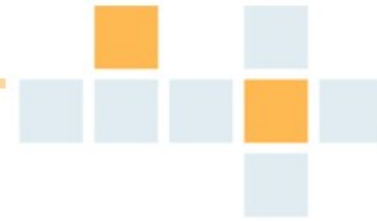
- Unit tests
- Functional tests
- Acceptance tests
- Integration tests
- Smoke tests
- Stress tests
- Corner case tests
- Long-running tests
- Manual tests
- Security tests
- Automatic UI tests
- Internationalization tests
- Multiplatform tests
- External tests

Optimizing SQ > Development process impacting quality



- Formal inspections
 - Areas: requirements, design, estimations, test plan, project cost
- Joint Application Design
- Peer code reviews
- Design horizon
 - Short vs long term, narrow vs. wide range
- Continuous integration
 - Fast build time, fast release
- Preparation for troubleshooting
 - Logs, metrics, remote support, remote patch application
- Refactoring
- Minor bugs fixing
- Performance tuning

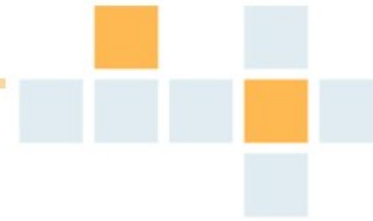
Product type, stage vs test types



| Product stage | Product type | Unit tests | Integration tests | Acceptance tests | Stress tests | Corner case tests | Long running tests | Manual tests | Security tests | Automated UI tests | Internationalization tests | Multi platform tests | External tests |
|------------------------------|--------------|------------|-------------------|------------------|--------------|-------------------|--------------------|--------------|------------------|--------------------|----------------------------|----------------------|----------------|
| POC | Any | could | won't | should | won't | won't | won't | should (1) | won't | won't | won't | won't | won't |
| MVP | Any | should | could | should | won't | won't | won't | should (1) | won't | won't | won't | won't | won't |
| Beta | Any | must | should | should | could | could | could | must (1) | should/ must (2) | could (1) | must (2) | won't | won't |
| Production (limited release) | Non-critical | must | must | must | should | should | could | must (1) | should/ must (2) | could (1) | must (2) | could (2) | could |
| | Critical | must | must | must | must | must | must | must (1) | should/ must (2) | could (1) | must (2) | could (2) | could |
| Production (general release) | Any | must | must | must | must | must | must | must (1) | should/ must (2) | could (1) | must (2) | should (2) | could |

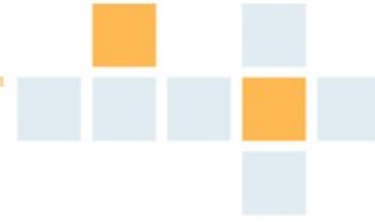
- (1) – if UI is present
- (2) – if applicable and required by priorities of software attributes

Product type, stage vs development standards



| Product stage | Product type | Formal inspections | Joint application design | In-depth code reviews | Broad design horizon | Continuous integration | Preparation for troubleshooting | Refactoring | Minor bugs fixing | Performance tuning |
|------------------------------|--------------|--------------------|--------------------------|-----------------------|----------------------|------------------------|---------------------------------|-------------|-------------------|--------------------|
| POC | Any | must | must | won't | won't | must | won't | won't | won't | won't |
| MVP | Any | must | must | could | won't | must | won't | could | won't | won't |
| Beta | Any | must | must | should | could | must | won't | should | could | could |
| Production (limited release) | Non-critical | must | must | must | could | must | should | must | should | should |
| | Critical | must | must | must | could | must | must | must | should | should |
| Production (general release) | Any | must | must | must | could | must | must | must | must | must |

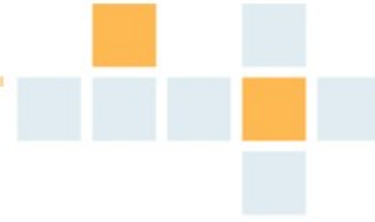
Example case study (ECS)



Machine learning solution

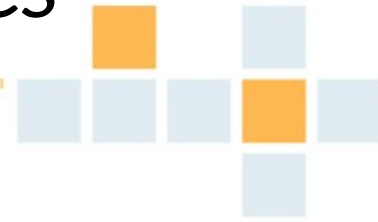
- Replacing house keys with face recognition algorithms
- Domestic market only
- Project type classification: safety-critical

ECS > ISO-25010 attributes priorities (production version)



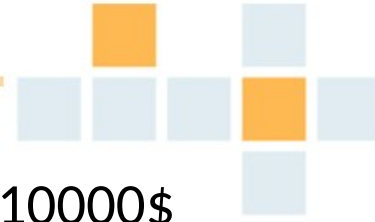
- Functional stability: high
- Performance efficiency: high
- Compatibility: low
- Usability: high
- Reliability: high
- Security: high
- Maintainability: medium
- Portability: low

ECS > Some stories for ISO-25010 attributes



- Functional stability
 - Need work well independent of:
 - Weather (sunny vs rainy)
 - Time of day (morning vs evening)
 - Seasons (summer light vs winter light)
 - Apparel changes (new glasses, huts, scarfs, ...)
 - Skin changes (scars, make up changes, ...)
 - Body structure changes (getting old, gaining weight, ..)
 - Mood changes (angry, sad face)
- Security
 - Impostor detection (owner's photo in front of camera)

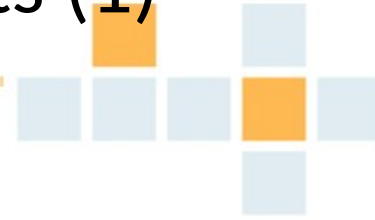
ECS > ISO-25010 attributes priorities mapped to development story points (production version)



- Initial estimation (in story points – one story point may relate to 10000\$ cost):
 - Functional stability: 100
 - Performance efficiency: 100
 - Usability: 30
 - Reliability: 170
 - Security: 200
- **TOTAL STORY POINTS: 600**

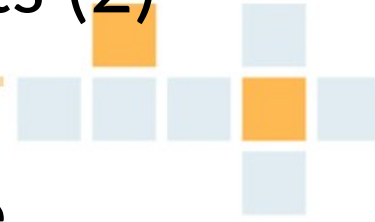
However, there is high risk that the requested quality will never be achieved and that the project will end up in failure.

Example case study (ECS): Relaxed requirements (1)



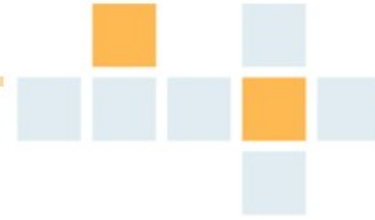
- Fingerprints to replace keys
 - Much simpler
 - Gigantic cost reduction
 - Still difficult to achieve 100% correctness and reliability

Example case study (ECS): Relaxed requirements (2)



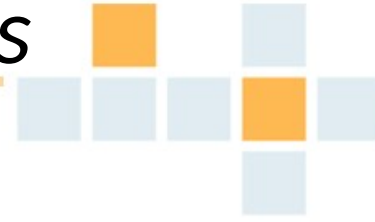
- Other idea: traditional keys still as an alternative
 - Software failures acceptable
 - Project type classification: now non-critical

ECS > Project stages



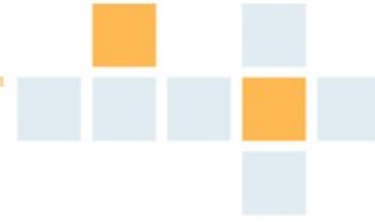
- Project stages
 - POC: verification of algorithms (will it work?)
 - MVP: a website allowing to open virtual house with face recognition; purchase request possible (is it needed?)
 - Beta: working software with cameras and locks, limited customers
 - Production

ECS > New ISO-25010 attributes priorities (production version), relaxed requirements



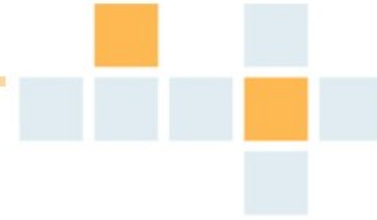
- Functional stability: medium
- Performance efficiency: high
- Compatibility: low
- Usability: high
- Reliability: medium
- Security: medium
- Maintainability: medium
- Portability: low

ECS > New ISO-25010 subattributes priorities (production version)



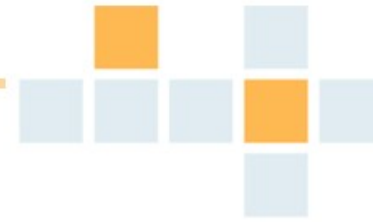
- Functional stability
 - correctness: high for positives, medium for negatives
- Performance efficiency
 - time behavior: high
- Usability:
 - learnability: high
 - user error protection: medium
 - user interface aesthetics: medium
- Reliability
 - availability: medium
- Security
 - authenticity: high

ECS > New ISO-25010 attributes priorities mapped to development story points (production version)



- Functional stability
 - correctness: 40
- Performance efficiency
 - time behavior: 20
- Usability:
 - learnability: 10
 - user error protection: 10
 - user interface aesthetics: 10
- Reliability
 - availability: 40
- Security
 - authenticity: 20
- **TOTAL STORY POINTS: 150 (cost reduction: 4 x)**

ECS > ISO-25010 attributes priorities mapped to development story points by stage

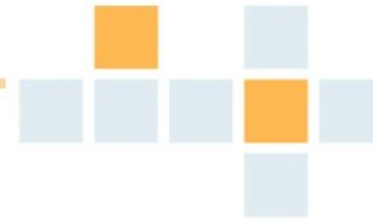


| Product stage | correctness | time behavior | learnability | user error protection | aesthetics | availability | authenticity |
|------------------------------|-------------|---------------|--------------|-----------------------|------------|--------------|--------------|
| POC | 5 | 2 | - | - | - | - | 4 |
| MVP | 5 | 2 | 2 | - | 2 | - | 4 |
| Beta | 10 | 8 | 4 | 4 | 4 | 15 | 4 |
| Production (limited release) | 10 | 4 | 2 | 4 | 2 | 10 | 4 |
| Production (general release) | 10 | 4 | 2 | 2 | 2 | 15 | 4 |

Total story points:

- POC: 11
- MVP: 15 (cancelling the product possible at lower cost)
- Beta: 49
- Production (limited release): 36
- Production (general release): 39

ECS > Product type, stage vs test types – story points

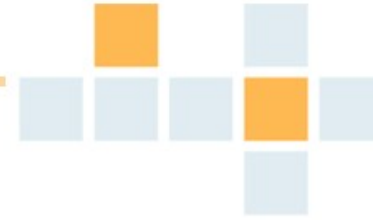


| Product stage | Unit tests | Integration tests | Acceptance tests | Stress tests | Corner case tests | Long running tests | Manual tests | Security tests | Automated UI tests | Inter-nationalization tests | Multi platform tests | External tests |
|------------------------------|------------|-------------------|------------------|--------------|-------------------|--------------------|--------------|----------------|--------------------|-----------------------------|----------------------|----------------|
| POC | - | - | 1 | - | - | - | 1 | - | - | - | - | - |
| MVP | 1 | 1 | 2 | - | - | - | 1 | - | - | - | - | - |
| Beta | 1 | 2 | 3 | - | 1 | - | 2 | 2 | - | - | - | - |
| Production (limited release) | 1 | 2 | 5 | - | 3 | 2 | 3 | 2 | - | - | - | - |
| Production (general release) | 1 | 2 | 5 | - | 3 | 4 | 3 | 2 | - | - | - | 5 |

Total story points:

- POC: 2
- MVP: 5
- Beta: 11
- Production (limited release): 18
- Production (general release): 25

ECS > Product type, stage vs development standards selection - story points

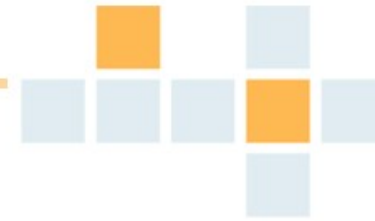


| Product stage | Formal inspections | Joint application design | In-depth code reviews | Broad design horizon | Continuous integration | Preparation for troubleshooting | Refactoring | Minor bugs fixing | Performance tuning |
|------------------------------|--------------------|--------------------------|-----------------------|----------------------|------------------------|---------------------------------|-------------|-------------------|--------------------|
| POC | 1 | 1 | - | - | 1 | - | - | - | - |
| MVP | 1 | 1 | - | - | 1 | - | - | - | - |
| Beta | 1 | 1 | 1 | - | 2 | - | 2 | - | - |
| Production (limited release) | 1 | 1 | 1 | - | 2 | 2 | 2 | 2 | - |
| Production (general release) | 1 | 1 | 1 | - | 2 | 2 | 2 | 2 | - |

Total story points:

- POC: 3
- MVP: 3
- Beta: 7
- Production (limited release): 11
- Production (general release): 11

ECS > Example influence of insufficient unit testing (additional story points)

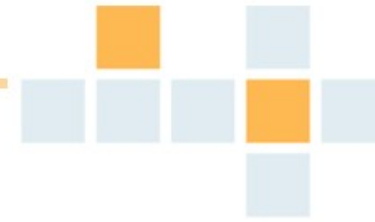


| Product stage | correctness | time behavior | learnability | user error protection | aesthetics | availability | authenticity |
|------------------------------|-------------|---------------|--------------|-----------------------|------------|--------------|--------------|
| POC | 5 + 1 | 2 + 1 | - | - | - | - | 4 + 1 |
| MVP | 5 + 1 | 2 + 1 | 2 + 1 | - | 2 | - | 4 + 1 |
| Beta | 10 + 10 | 8 + 8 | 4 + 4 | 4 + 4 | 4 + 4 | 15 + 15 | 4 + 4 |
| Production (limited release) | 10 + 10 | 4 + 4 | 2 + 2 | 4 + 4 | 2 + 2 | 10 + 10 | 4 + 4 |
| Production (general release) | 10 + 10 | 4 + 4 | 2 + 2 | 2 + 4 | 2 + 2 | 15 + 15 | 4 + 4 |

Reasons

- More bugs
- Slow feedback loop
- Code hard to test

ECS > Example influence of insufficient formal inspections and joint application design (additional story points)

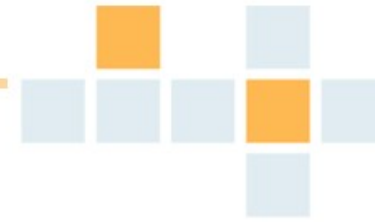


| Product stage | correctness | time behavior | learnability | user error protection | aesthetics | availability | authenticity |
|------------------------------|-------------|---------------|--------------|-----------------------|------------|--------------|--------------|
| POC | 5 + 2 | 2 + 1 | - | - | - | - | 4 + 1 |
| MVP | 5 + 5 | 2 + 2 | 2 + 1 | - | 2 | - | 4 + 2 |
| Beta | 10 + 8 | 8 + 5 | 4 + 2 | 4 + 2 | 4 + 1 | 15 + 5 | 4 + 1 |
| Production (limited release) | 10 + 6 | 4 + 2 | 2 + 2 | 4 + 2 | 2 + 1 | 10 + 5 | 4 + 1 |
| Production (general release) | 10 + 5 | 4 + 2 | 2 + 2 | 2 + 2 | 2 + 1 | 15 + 5 | 4 + 1 |

Reasons

- Misdirected development effort
- Requirements misunderstanding
- Design defects

ECS > Example influence of insufficient continuous integration effort (additional story points)

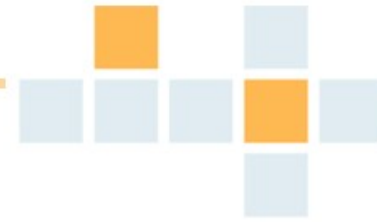


| Product stage | correctness | time behavior | learnability | user error protection | aesthetics | availability | authenticity |
|------------------------------|-------------|---------------|--------------|-----------------------|------------|--------------|--------------|
| POC | 5 + 1 | 2 + 1 | - | - | - | - | 4 + 1 |
| MVP | 5 + 1 | 2 + 1 | 2 + 1 | - | 2 | - | 4 + 1 |
| Beta | 10 + 2 | 8 + 2 | 4 + 1 | 4 + 1 | 4 + 1 | 15 + 2 | 4 + 1 |
| Production (limited release) | 10 + 2 | 4 + 1 | 2 + 1 | 4 + 1 | 2 + 1 | 10 + 2 | 4 + 1 |
| Production (general release) | 10 + 2 | 4 + 1 | 2 + 1 | 2 + 1 | 2 + 1 | 15 + 2 | 4 + 1 |

Reasons

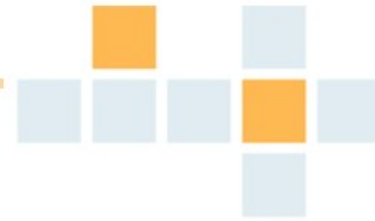
- Slower builds
- Slower code integration
- Longer feedback loop

About 9LivesData



- 9LivesData expertise
 - Enterprise solutions, including
 - storage products
 - machine learning solutions
 - Proof of Concepts
 - Minimum Viable Products
 - Productization of prototypes
- Contact
 - Cezary Dubnicki, PhD
 - projects@9livesdata.com

Summary



- Optimal software quality
 - High quality achieved in optimal way (time, money)
 - Never about making quality low
- Steps
 - Requirements negotiations
 - ISO-25010 quality attributes prioritization
 - Development effort direction
 - Project type and stage classification
 - Selection of appropriate test types
 - Development process tailoring

References



- Capers Jones, Olivier Bonsignour, *The Economics of Software Quality*
- Ivan Mistrik, *Software Quality Assurance: In Large Scale and Complex Software-intensive Systems*
- James Whittaker, Jason Arbon, Jeff Carollo, *How Google Tests Software*
- Kent Beck, *Test Driven Development: By Example*
- Ash Maurya, *Running Lean*
- Eric Ries, *The Lean Startup*
- Fred Brooks, *The Mythical Man-Month*
- Edward Yourdon, *Death March*
- Alan Cooper, *The Inmates Are Running the Asylum*