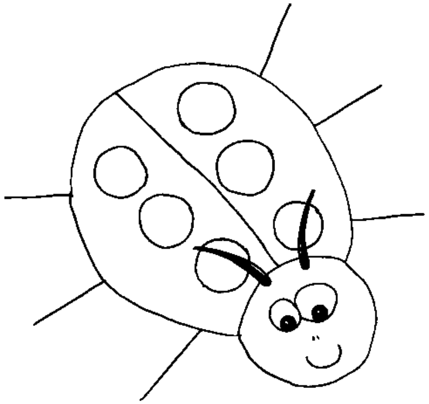


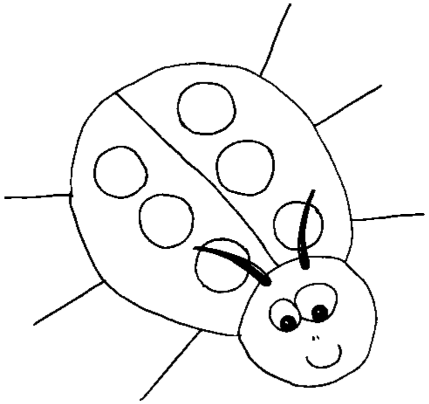
Test Ideas

Robert Sabourin

AmiBug.Com, Inc.

Montreal, Canada

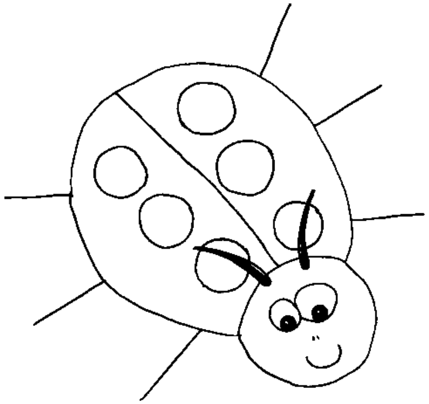
robsab@gmail.com



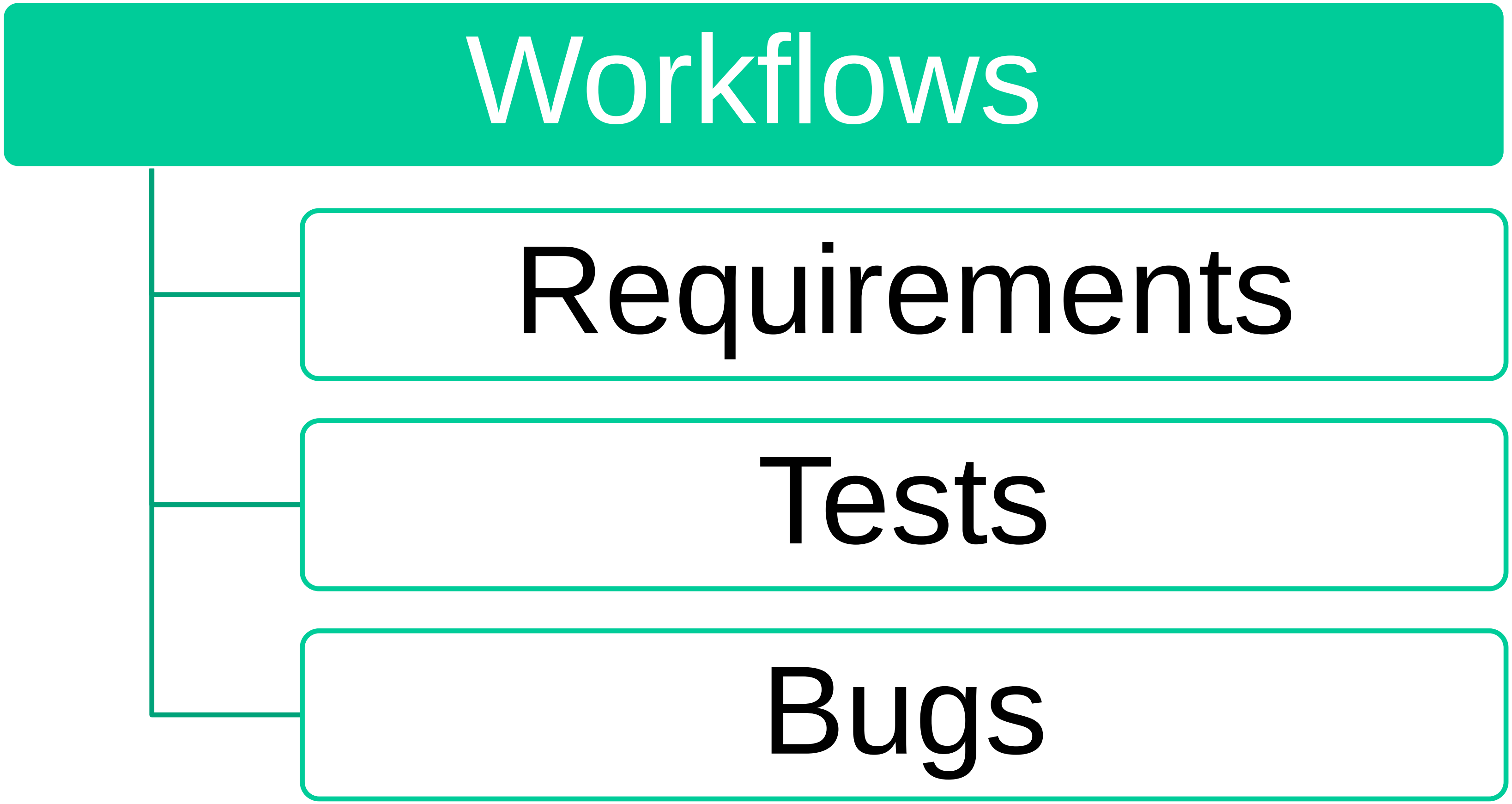
Just In Time Testing

Decision Making

Capture testing ideas



Decision Making



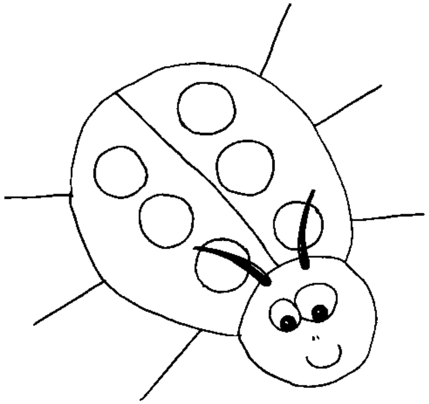
Workflows

Requirements

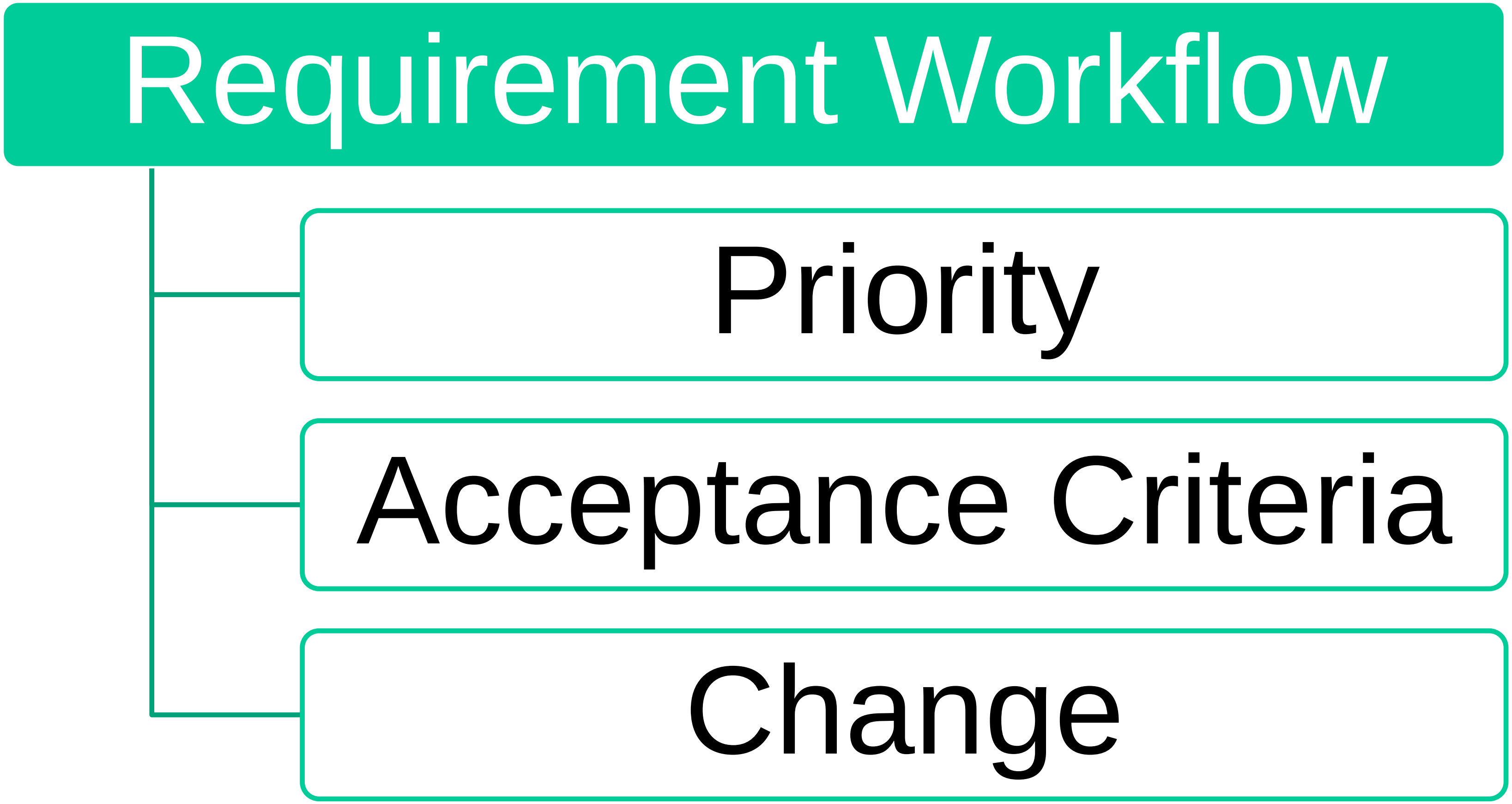
Tests

Bugs

Capture testing ideas



Decision Making



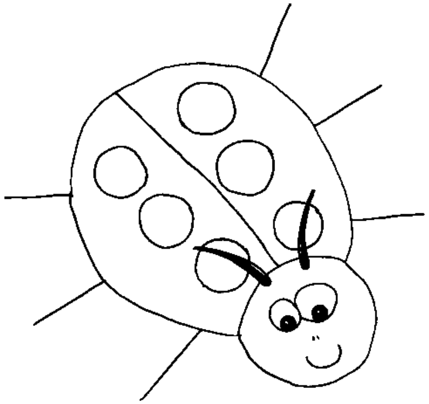
Requirement Workflow

Priority

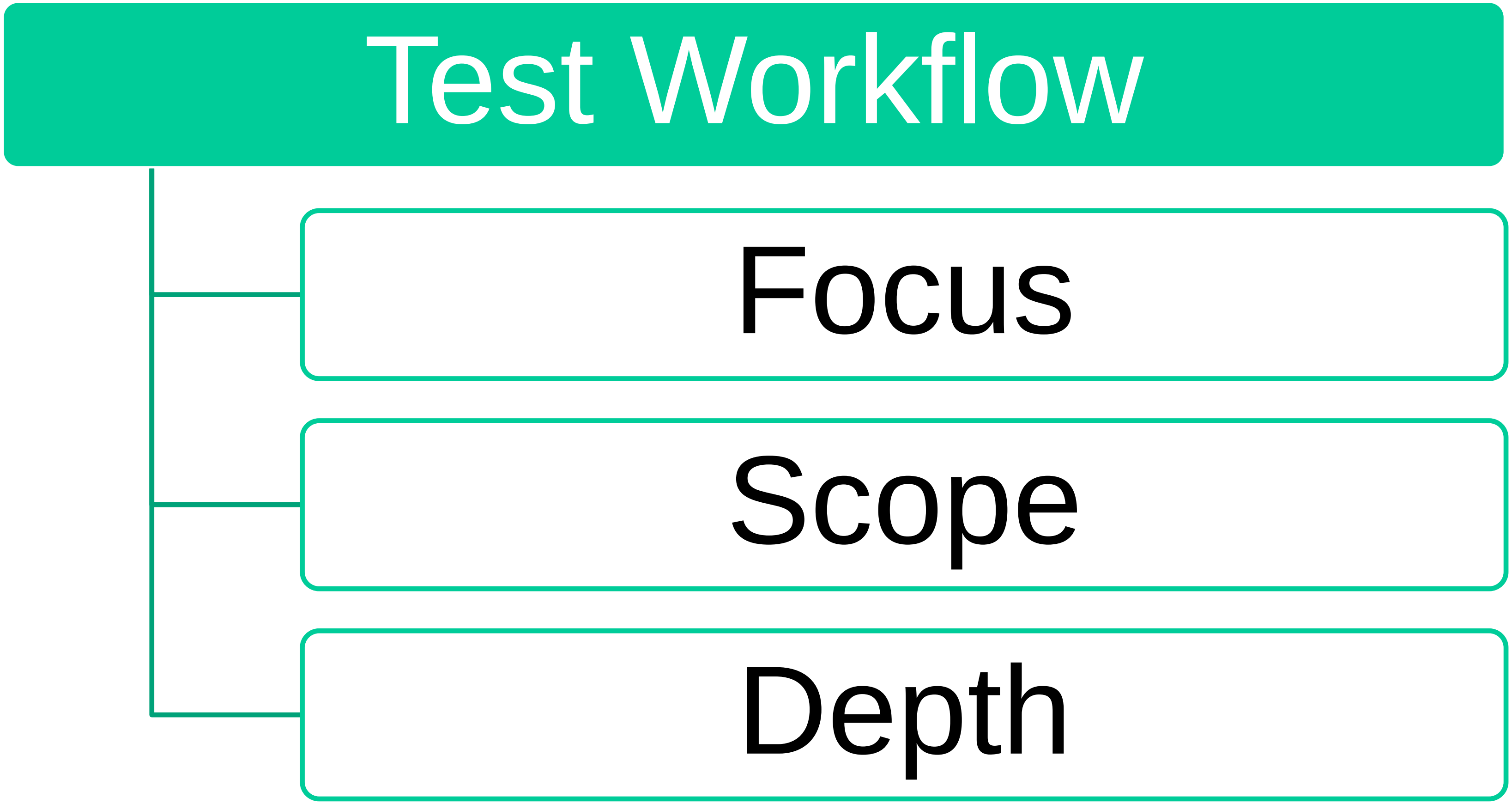
Acceptance Criteria

Change

Capture testing ideas



Decision Making



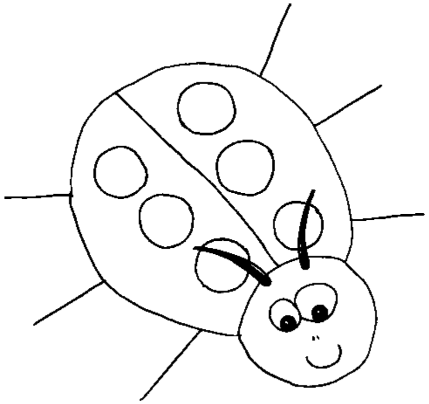
Test Workflow

Focus

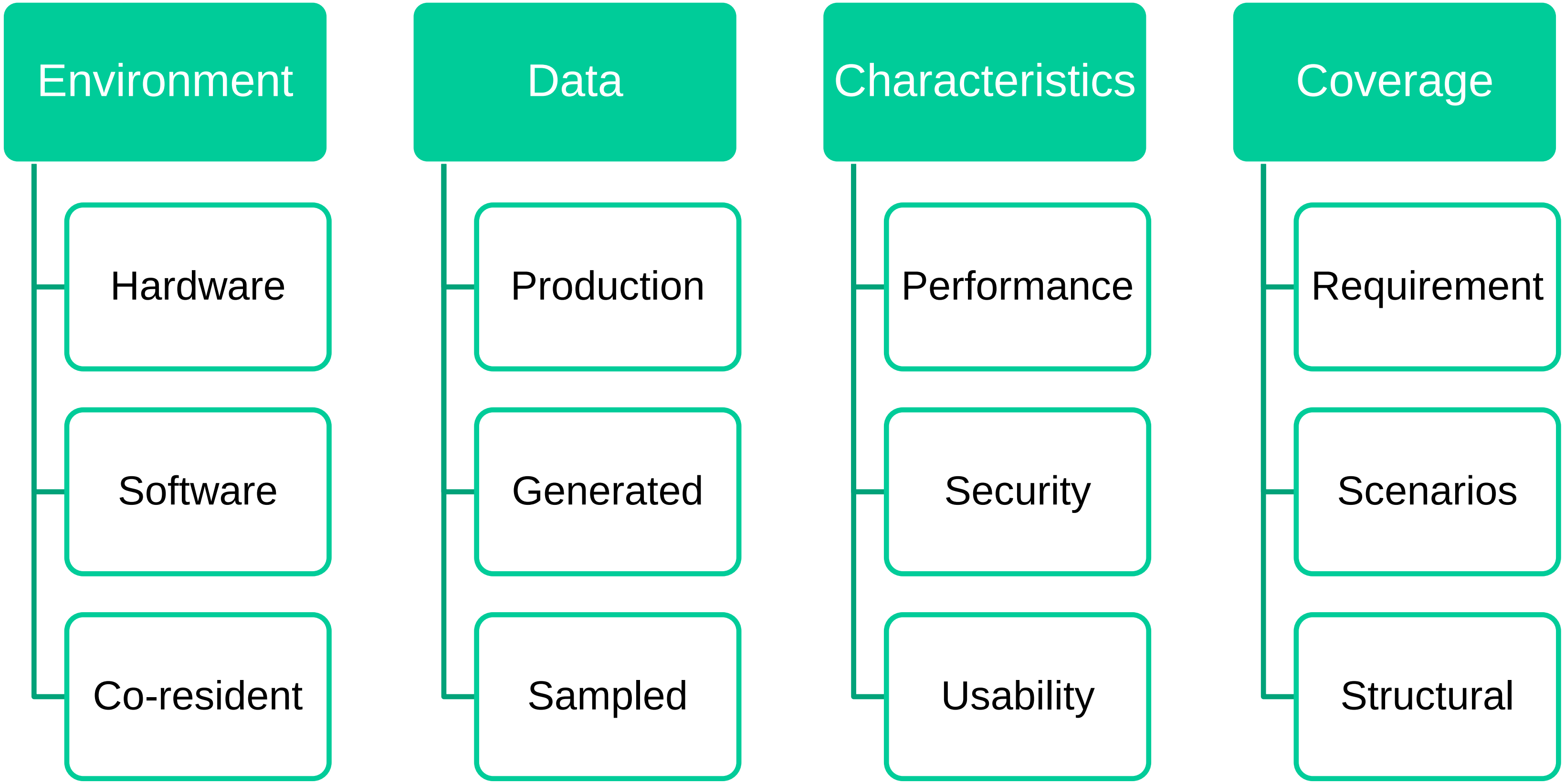
Scope

Depth

Capture testing ideas

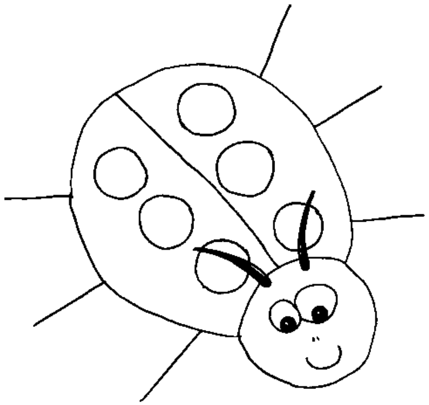


Scope of Testing

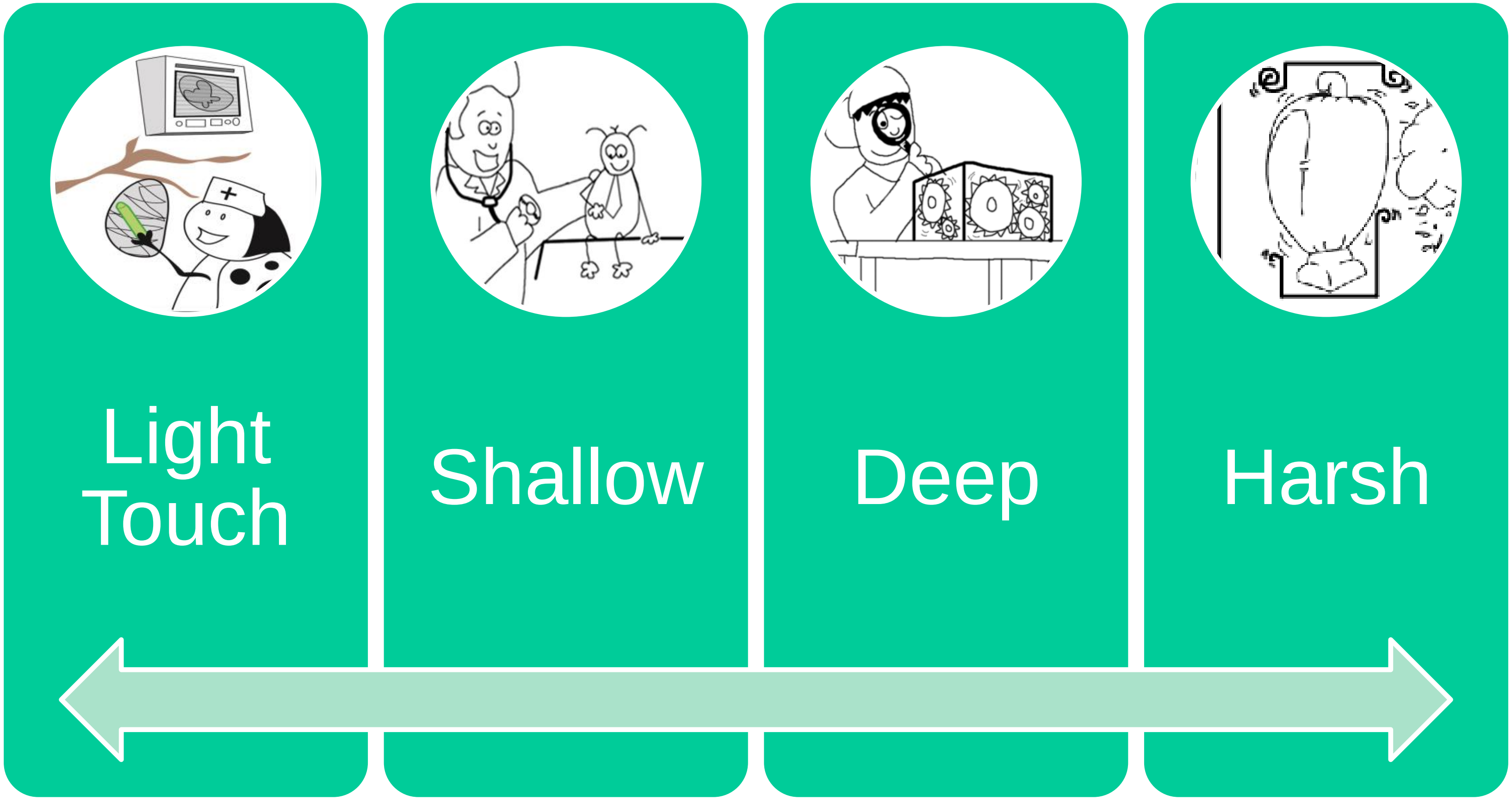


- Environment
  - Hardware
  - Software
  - Co-resident
- Data
  - Production
  - Generated
  - Sampled
- Characteristics
  - Performance
  - Security
  - Usability
- Coverage
  - Requirement
  - Scenarios
  - Structural

Capture testing ideas



Depth of Testing



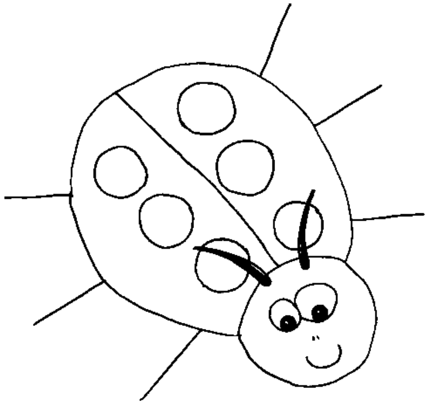
Light Touch

Shallow

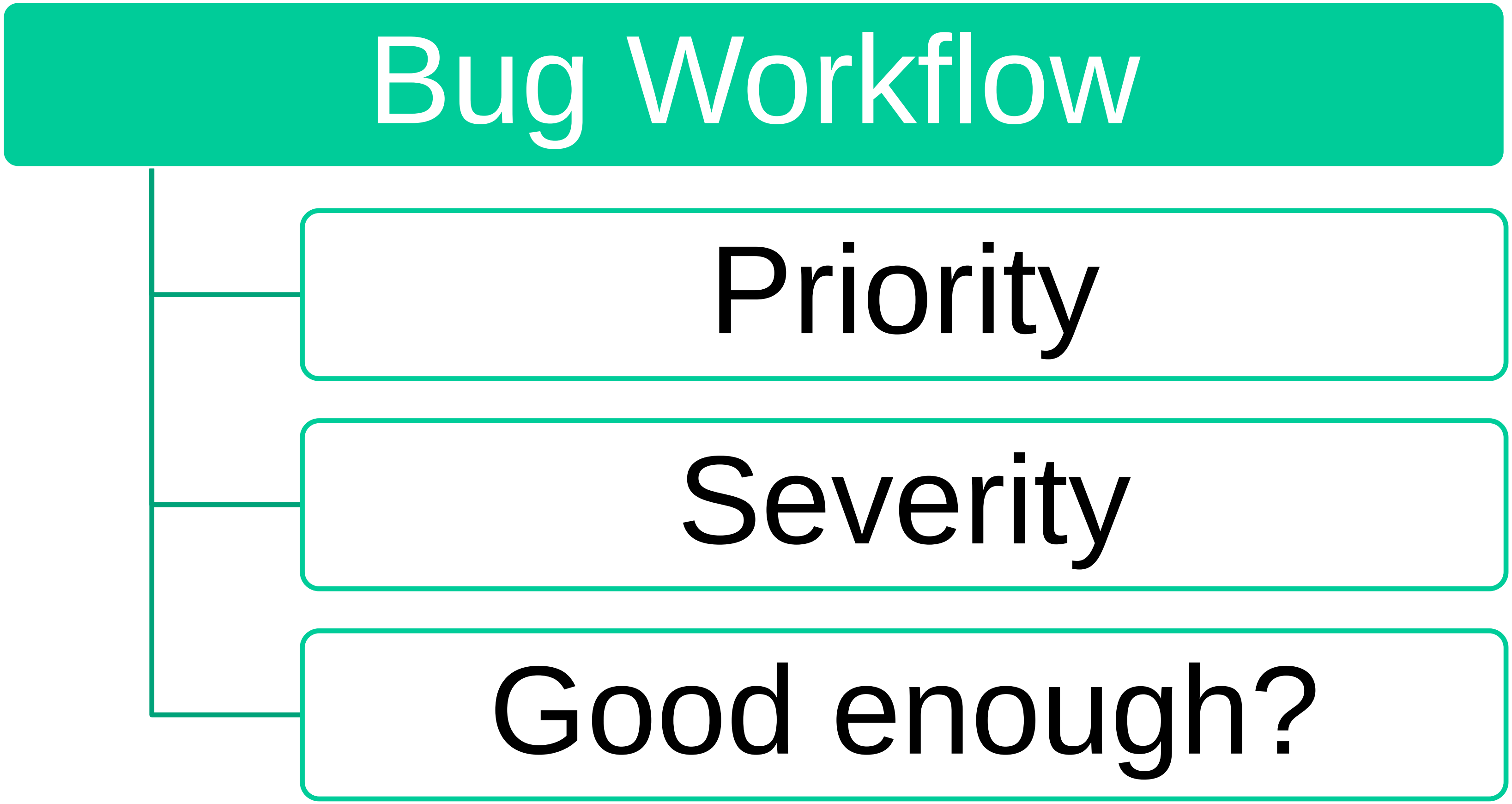
Deep

Harsh

Capture testing ideas



Decision Making



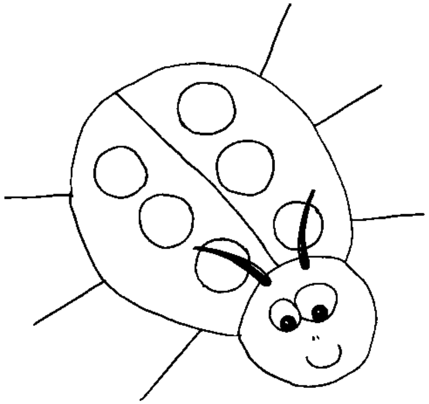
Bug Workflow

Priority

Severity

Good enough?

Capture testing ideas



Just In Time Testing

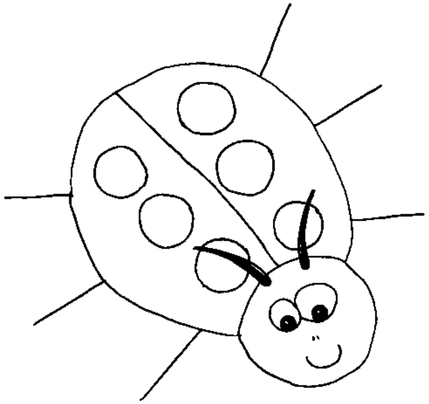
Test Triage

Capture testing ideas

Yoda

"No! Try not, Do. Or do not.

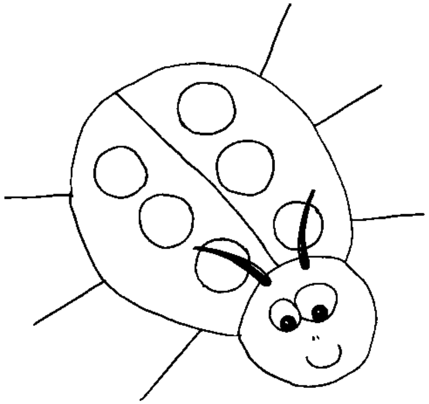
There is no try."



Testing Ideas

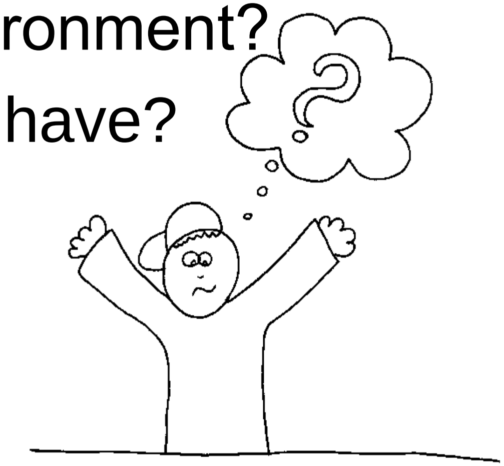
- Collect all testing ideas you can find!
  - – List
  - – Sort
  - – Organize
  - – Shuffle

Plan to support change

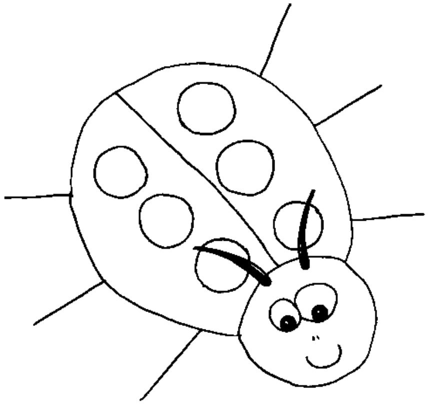


Testing Ideas

- How to find them?
  - Does system do what it is suppose to do?
  - Does the system do things it is not supposed to?
  - How can the system break?
  - How does the system react to it’s environment?
  - What characteristics must the system have?
  - Why have similar systems failed?
  - How have previous projects failed?



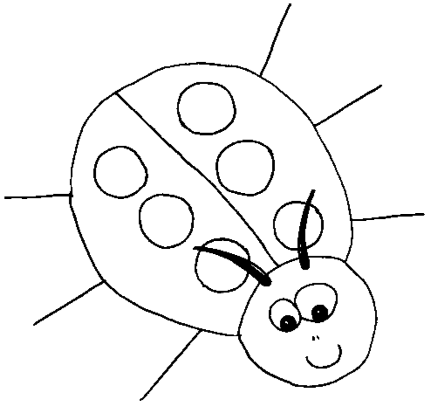
Plan to support change



Testing Ideas

- Collect testing ideas
- From testing ideas build a series of testing objectives
  - Each can be assigned *as work* to testers
  - Each can include *all, part of, or multiple testing ideas*

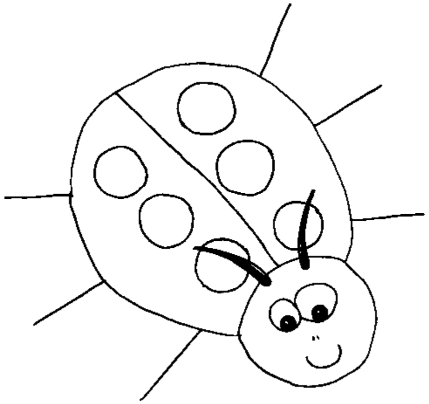
Capture testing ideas



Testing Ideas

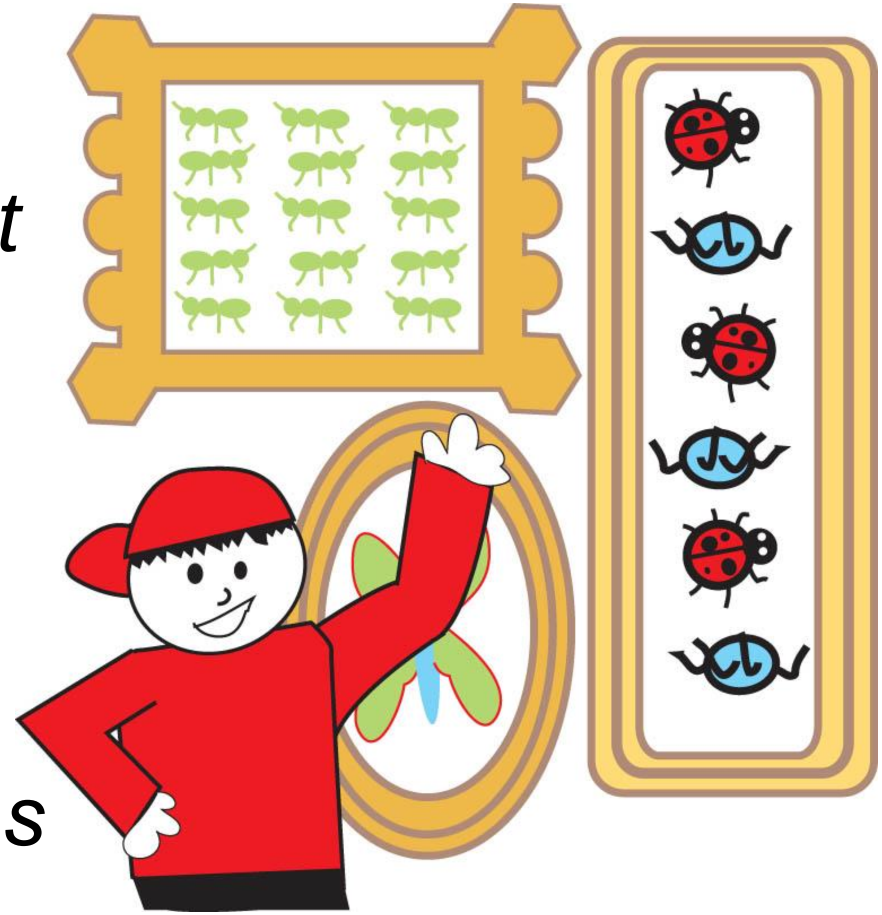
- I often use *Index Cards*
  - – Unique id
  - – One testing idea per card
  - – Colour indicates source
  - – Shuffled and reviewed
  - – Organized and reorganized
  - – Sorted, grouped, prioritized and collected

Capture testing ideas

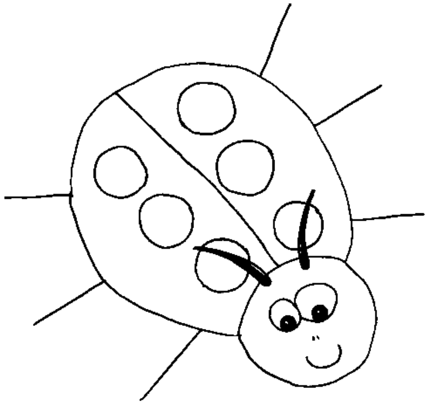


Testing Ideas

- I collect test ideas
- … *when I learn about the project*
- … *when planning*
- … *while testing*
- … *in production*
- … *from many sources*

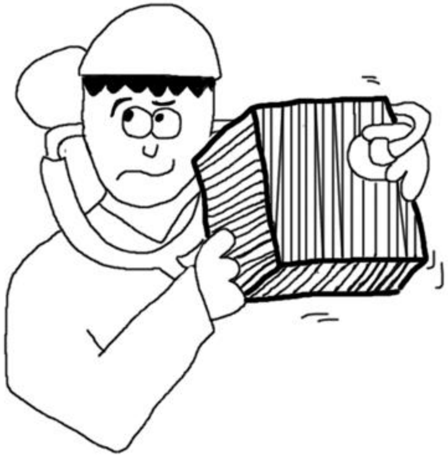


Capture testing ideas



Test Idea Sources

Capabilities	Failure Modes
Quality Factors	Usage Scenarios
Creative Ideas	States
Data	Environments
White Box	Taxonomies
Across Story Relationships	Software Breaking
End to End Testing	Sequencing

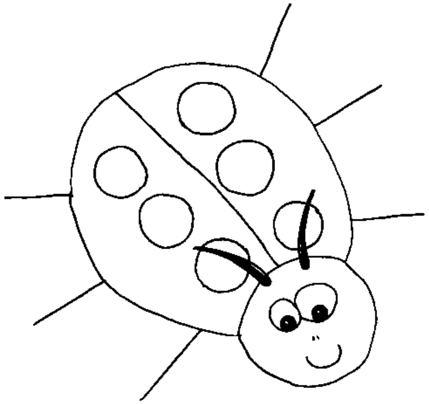


Capture testing ideas

Testing Ideas

- Reconnaissance
  - We become truffle snorting pigs and try to find useful information in all evidence we discover
  - We can even get good ideas from <u>out of date</u> sources

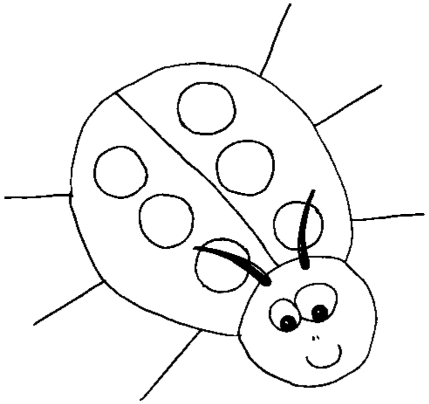
Capture testing ideas



Testing Ideas

- Capabilities
  - Use cases
  - Functional requirements
  - Documented requirements
  - Implicit requirements

Capture testing ideas



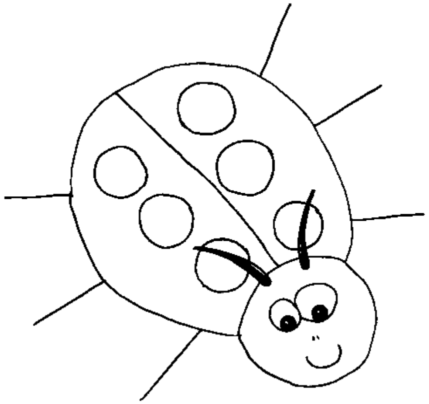
Testing Ideas

Capabilities

- Use cases
- Functional requirements
- Documented requirements
- Implicit requirements

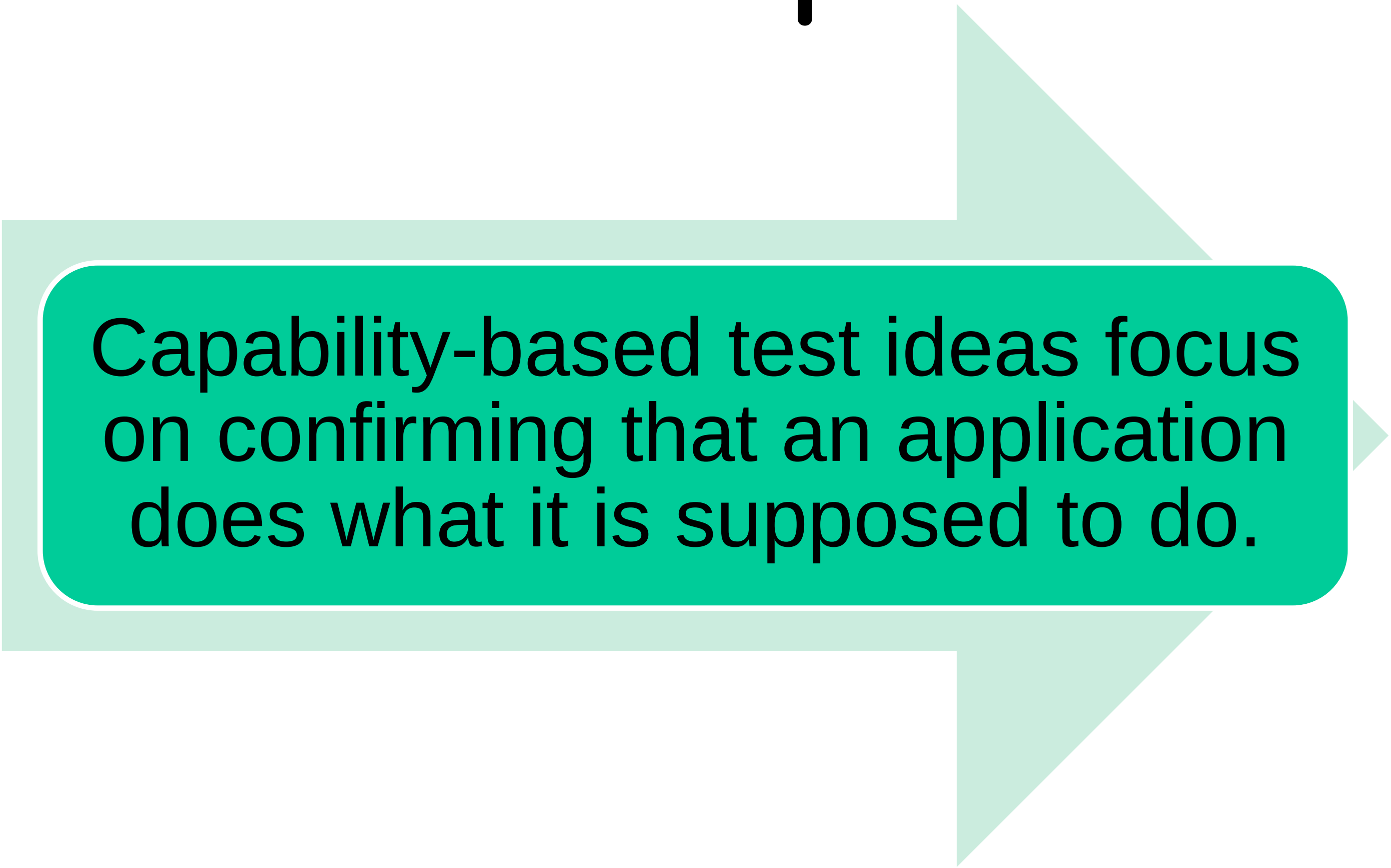
- *Does the system do what it is supposed to do?*

Capture testing ideas



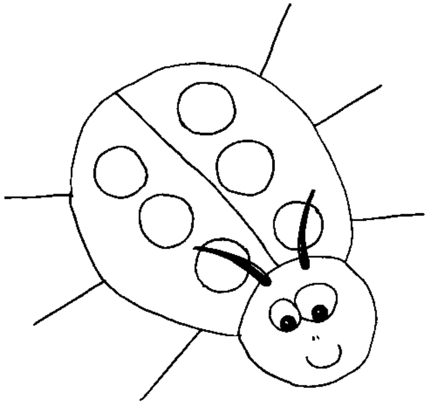
Testing Ideas

Capabilities



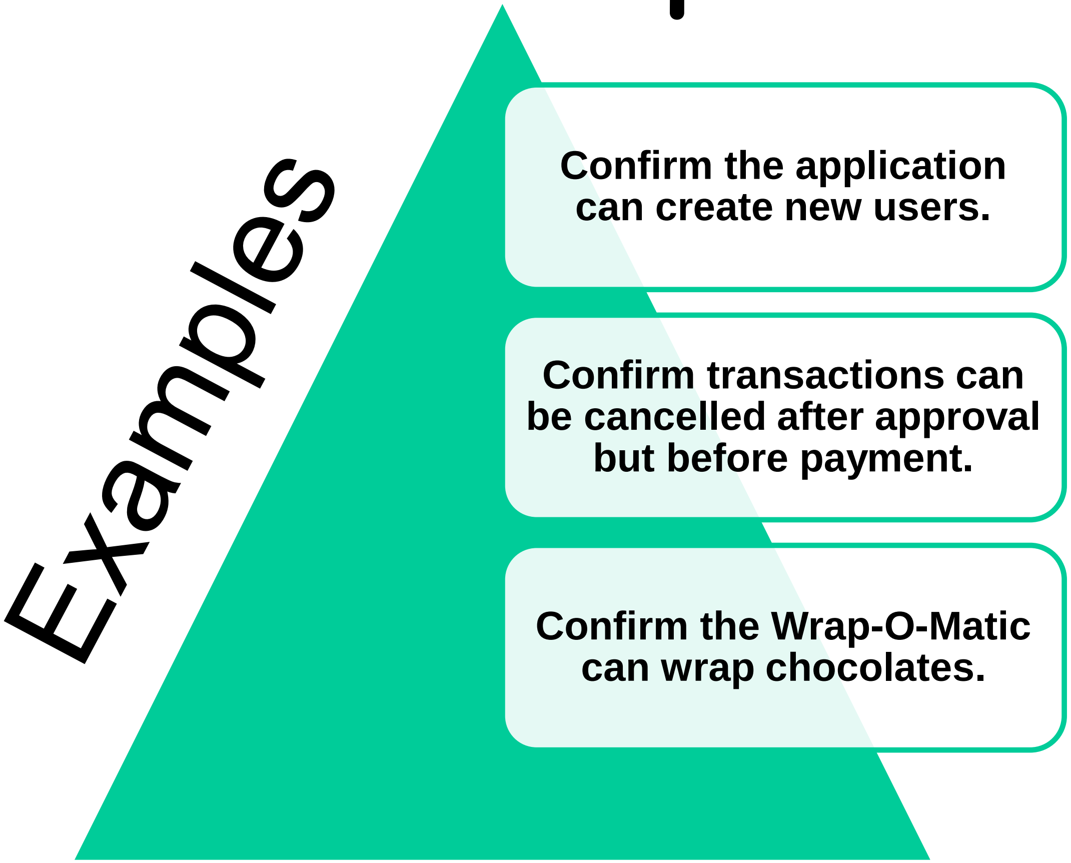
Capability-based test ideas focus on confirming that an application does what it is supposed to do.

Capture testing ideas



Testing Ideas

Capabilities

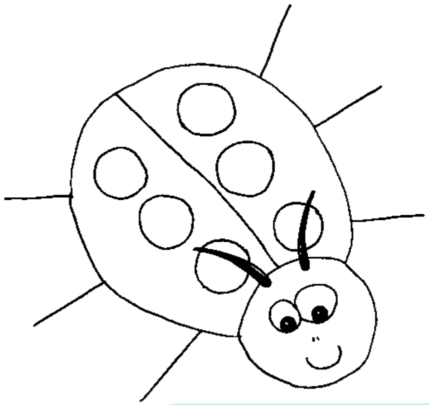


Confirm the application can create new users.

Confirm transactions can be cancelled after approval but before payment.

Confirm the Wrap-O-Matic can wrap chocolates.

Capture testing ideas



Testing Ideas

Sources

Capabilities

Documents

- Manuals
- Help Systems
- Packaging
- Emails
- Requirements
- Design
- Code
- Schemas

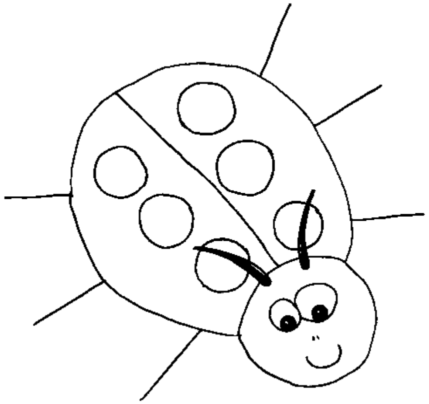
People

- Customers
- Users
- Sys Admin
- DBA
- Help Desk
- Sales
- Developers
- Testers

Exploration

- Reconnaissance
- Competition
- Old versions
- Gap Analysis
- Affinity Analysis
- Pair SME
- Pair Expert Users
- Pair Admin

Capture testing ideas



Testing Ideas

Capabilities

Capability Test Ideas

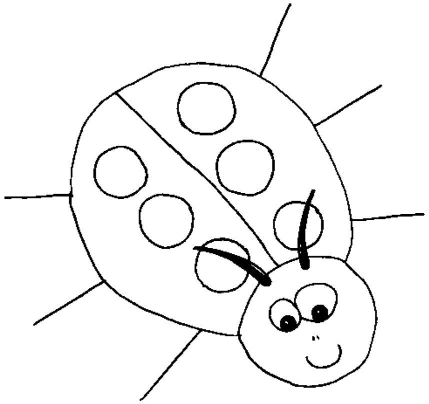
Find Objects

Identify Actions

Isolate Variables

Relate Objects, Actions & Variables

Capture testing ideas

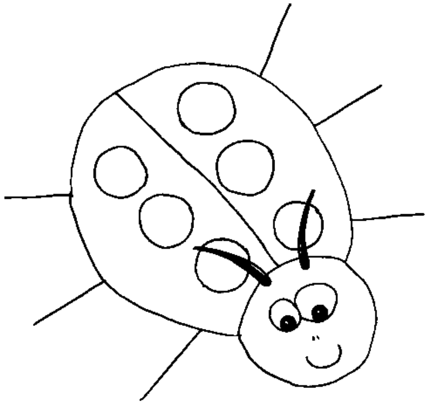


Testing Ideas Capabilities

Find Objects

- Testable objects are things I can learn about. Testable objects may be a process, screens, form, page, feature or function. They are often requirement “nouns”.

Capture testing ideas

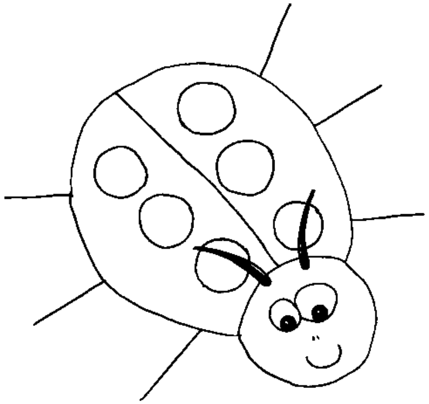


Testing Ideas
Capabilities

Identify Actions

- Once I have identified some testable objects I start hunting for actions. Actions are things that software does to testable objects. Actions are often “verbs” in requirement documents.

Capture testing ideas

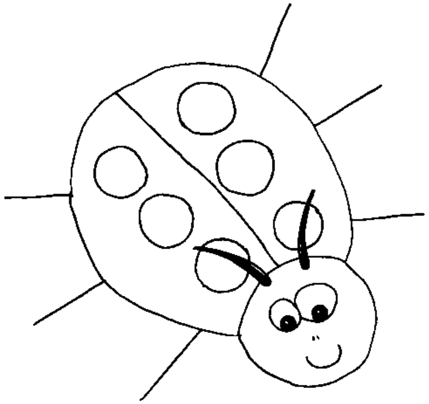


Testing Ideas Capabilities

Isolate Variables

- Having found testable objects and actions I search for variables. A variable is something which can change. The behaviour of software depends on the values variables take on.

Capture testing ideas

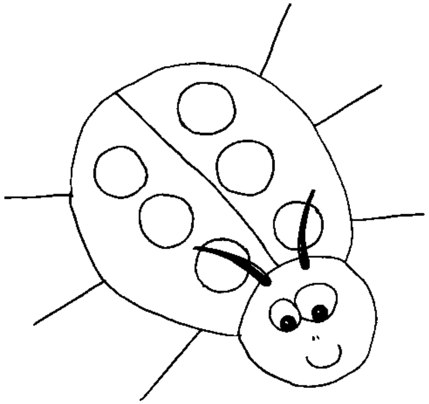


Testing Ideas Capabilities

Relate Them

- Capability test ideas explore the relationship between testable objects, actions and variables.

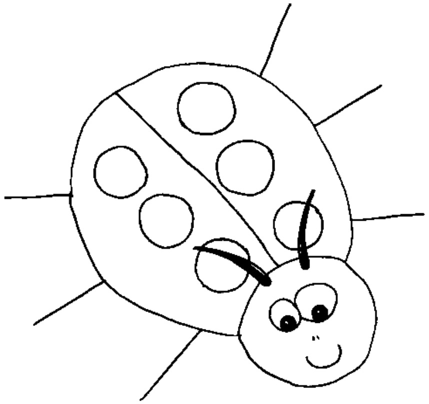
Capture testing ideas



Testing Ideas

- Failure Modes
  - What can break?
  - Reaction to invalid input?
  - How does software behave in constrained environment?
    - Memory
    - Disk Space
    - Network Bandwidth
    - CPU capacity
    - Shared resources
  - Stress, Load, Volume

Capture testing ideas

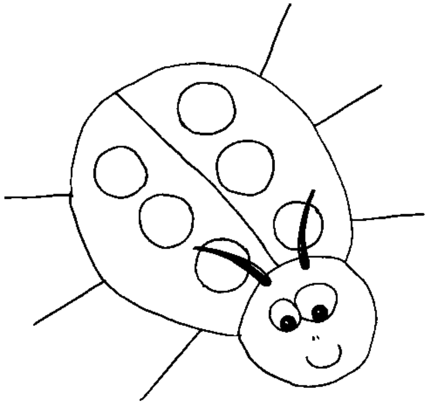


Testing Ideas
Failure Modes

Failure Modes Defined

- Failure mode test ideas focus on learning about the applications behaviour when something goes wrong.
- Failure mode test ideas are "what if" questions. They are often inspired by how a system is designed.
- I look at the objects, components, processes and interfaces in a system and then I ask myself, "what if they break?" or "what if they exhibit some sort of unanticipated failure?"

Capture testing ideas

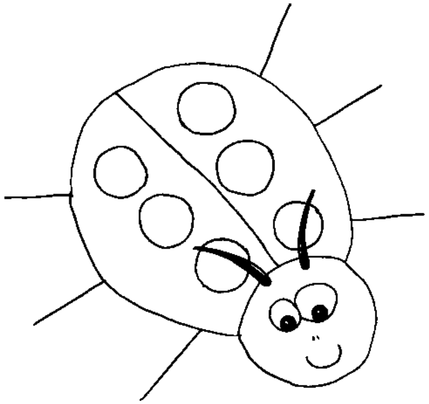


Testing Ideas
Failure Modes

Some Failure Mode Test Ideas

- What if the system does not have enough memory?
- What if the Wrap-O-Matic inbound conveyor belt breaks during a production run?
- What if the Wrap-O-Matic runs out of paper?
- What if an inbound ribbon spool breaks?

Capture testing ideas

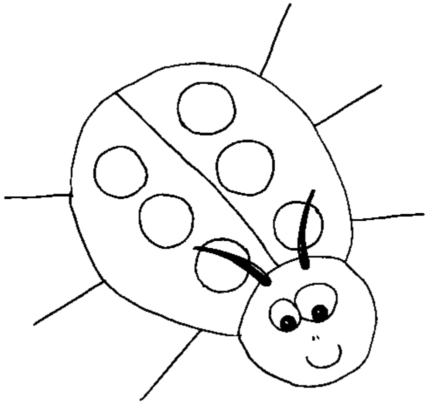


Testing Ideas
Failure Modes

Examples Failure Mode Test Ideas

- What if the rejected chocolate bin overflows?
- What if the rates of arrival of chocolates inbound exceeds the rate of Wrap-O-Matic production?
- What if typical production runs occur at low temperatures
- What if typical production runs occur at high temperatures

Capture testing ideas

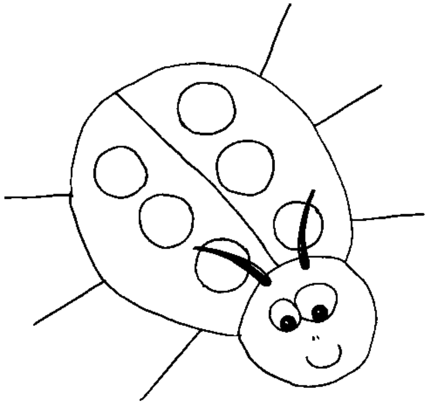


Testing Ideas
Failure Modes

Examples Failure Mode Test Ideas

- What if the Wrap-O-Matic operates for 48 hours non-stop
- How long can the Wrap-O-Matic operate with the hardest wrapping scenario without failure or need of maintenance

Capture testing ideas

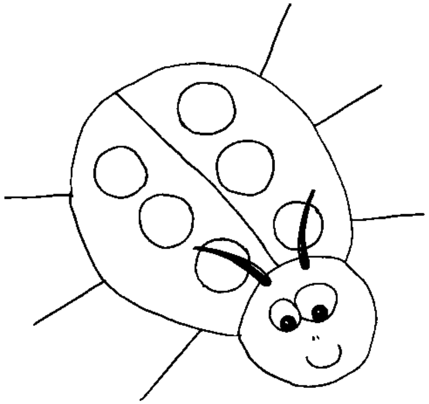


Testing Ideas
Failure Modes

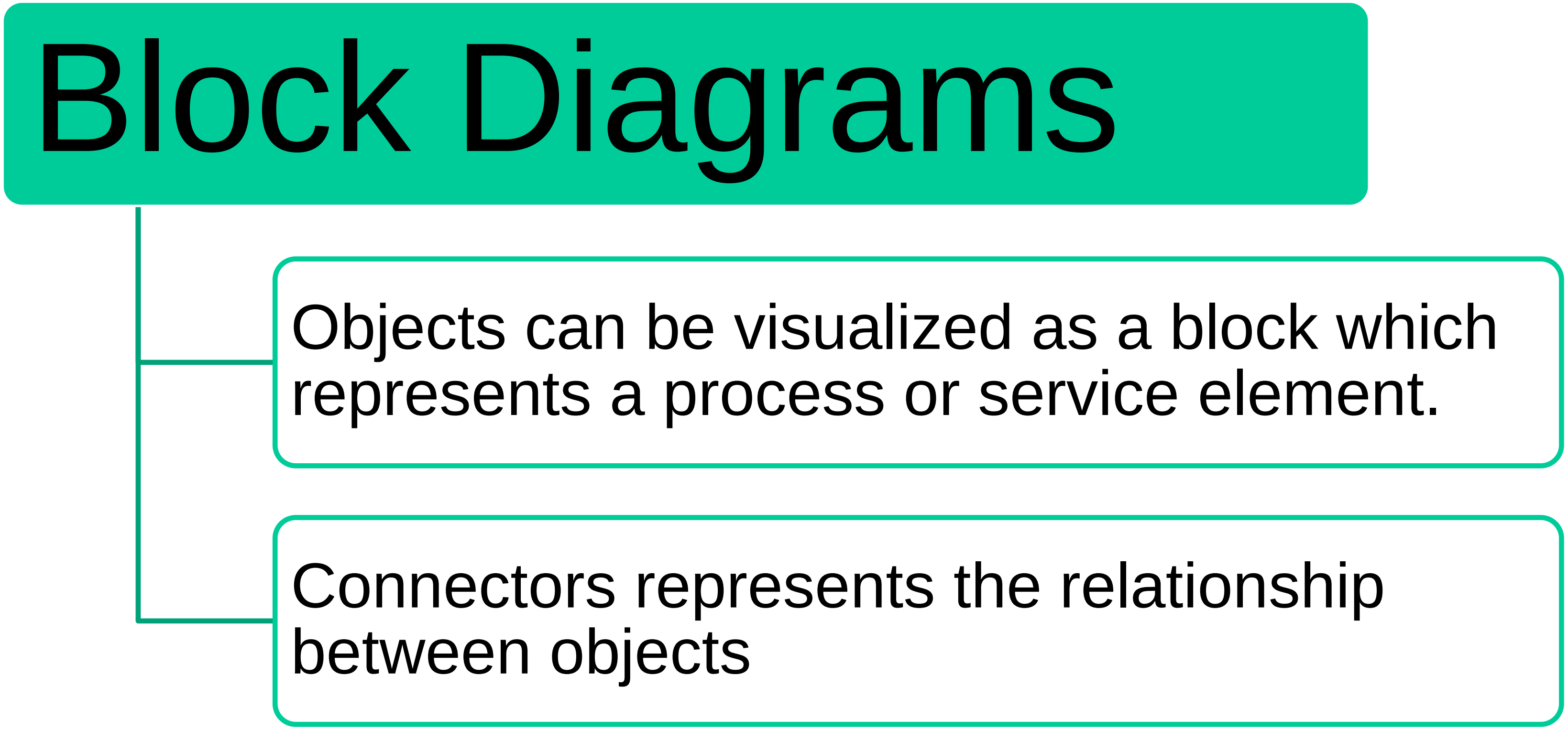
Block Diagrams

- Software Architecture. Foundations, Theory and Practise by Taylor, Medvidovic and Dasgify a treatment of software design artifacts identifies that software designs can be described in terms of objects and connectors.

Capture testing ideas



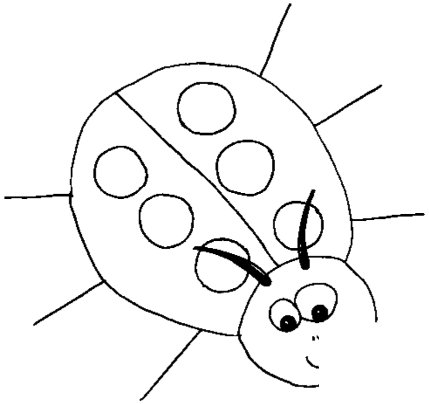
Testing Ideas
Failure Modes



Block Diagrams

- Objects can be visualized as a block which represents a process or service element.
- Connectors represents the relationship between objects

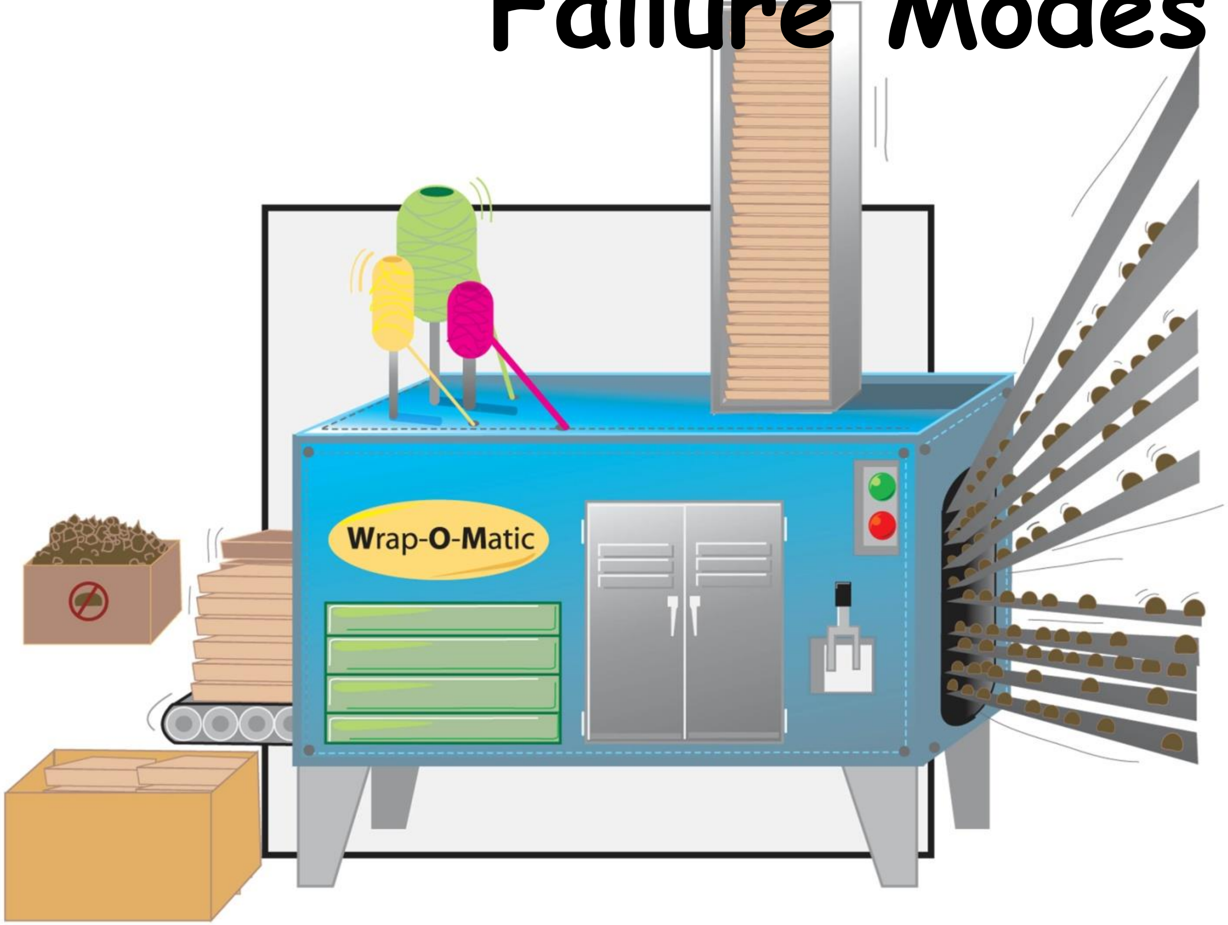
Capture testing ideas

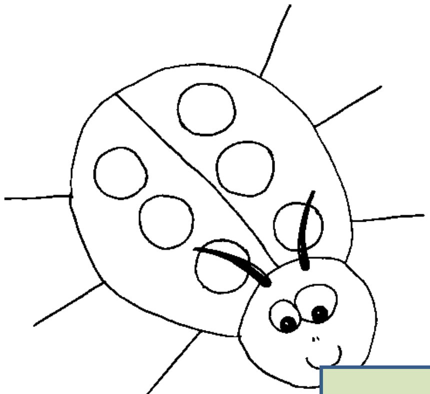


Testing Ideas
Failure Modes

Wrap-O-Matic

Capture testing ideas

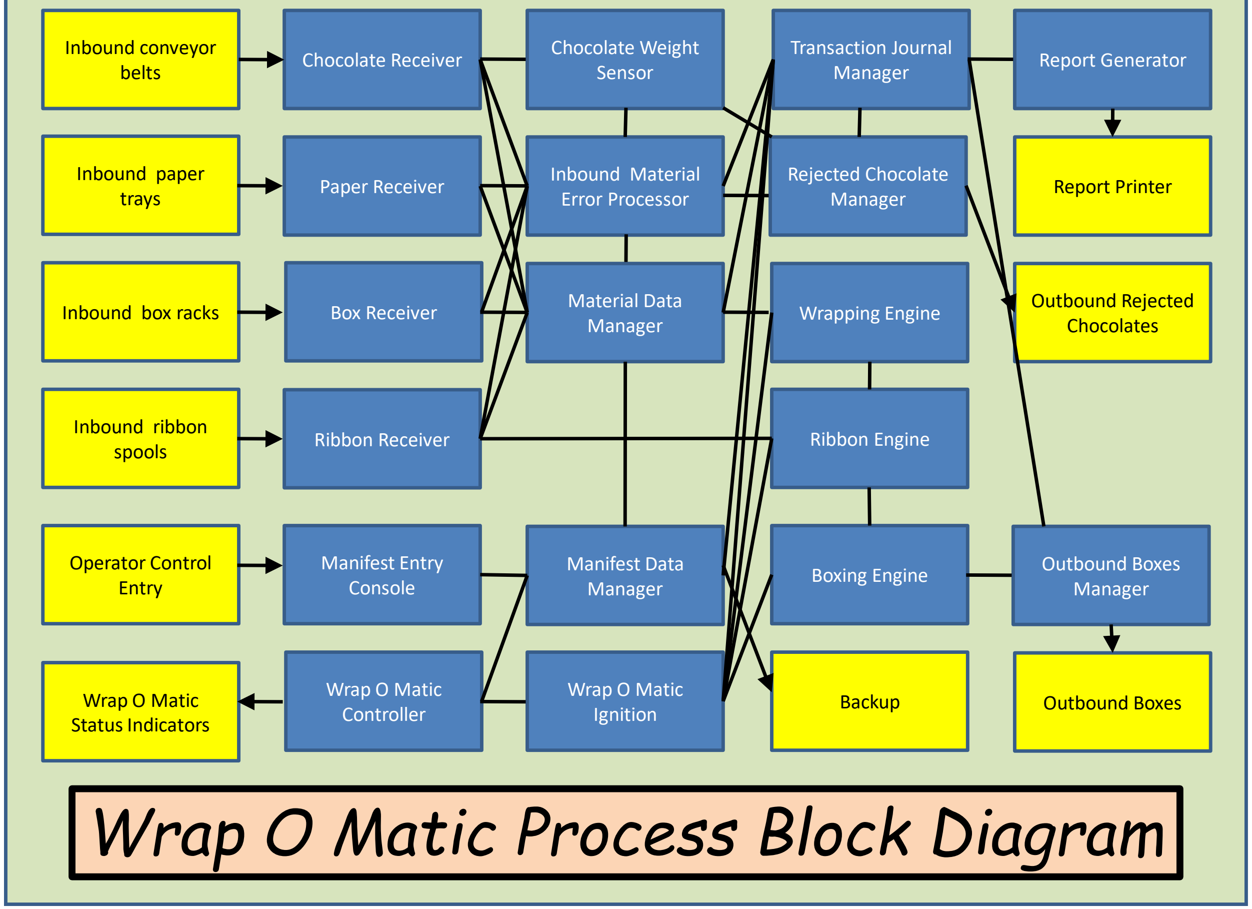




Testing Ideas
Failure Modes

Capture testing ideas

Block Diagram



Inbound conveyor belts

Chocolate Receiver

Chocolate Weight Sensor

Transaction Journal Manager

Report Generator

Inbound paper trays

Paper Receiver

Inbound Material Error Processor

Rejected Chocolate Manager

Report Printer

Inbound box racks

Box Receiver

Material Data Manager

Wrapping Engine

Outbound Rejected Chocolates

Inbound ribbon spools

Ribbon Receiver

Ribbon Engine

Operator Control Entry

Manifest Entry Console

Manifest Data Manager

Boxing Engine

Outbound Boxes Manager

Wrap O Matic Status Indicators

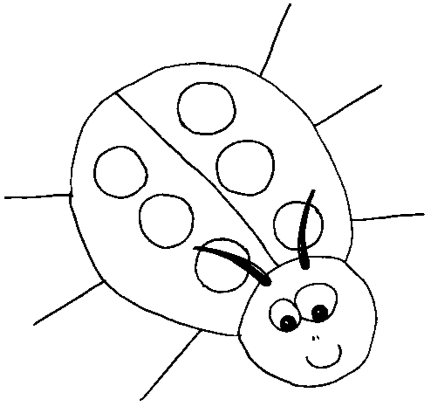
Wrap O Matic Controller

Wrap O Matic Ignition

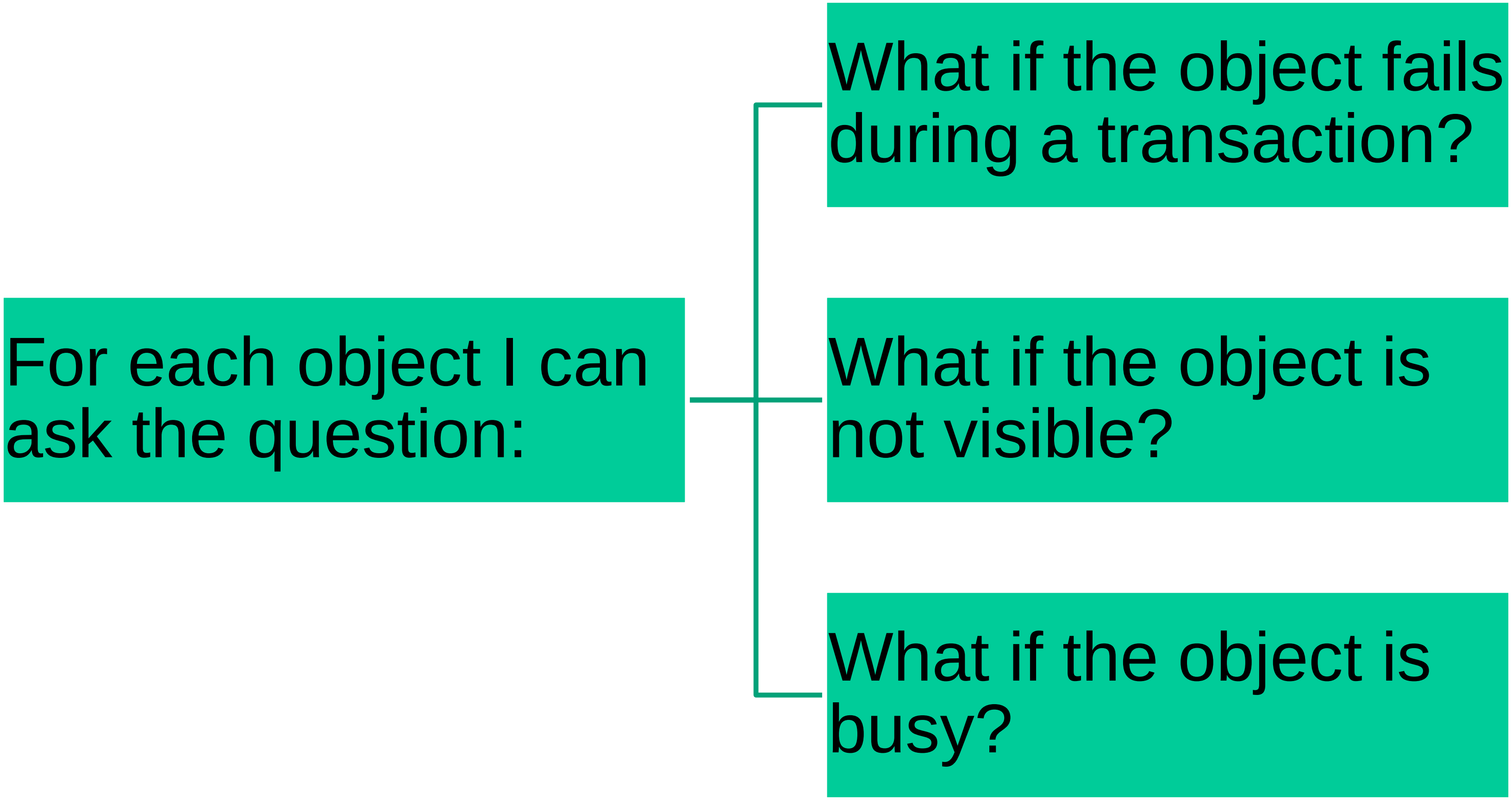
Backup

Outbound Boxes

Wrap O Matic Process Block Diagram



Testing Ideas
Failure Modes



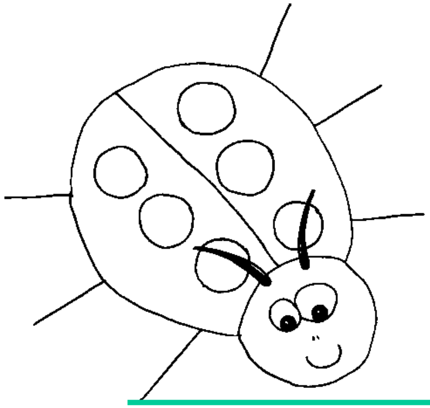
For each object I can ask the question:

What if the object fails during a transaction?

What if the object is not visible?

What if the object is busy?

Capture testing ideas



Testing Ideas
Failure Modes

Connector Procedure call

What if data transfer is by value instead of by reference?

What if Order of parameters is incorrect?

What if version of objects are out of sync?

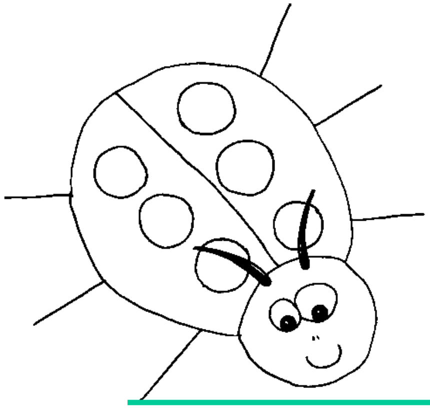
What if return value indicates an error?

What if order of execution is changed?

What if procedure is called from wrong object?

What if multiple threads use the procedure concurrently?

Capture testing ideas



Testing Ideas
Failure Modes

Connector Events

Which events trigger this connection?

Are events prioritized?

What if all events are high priority?

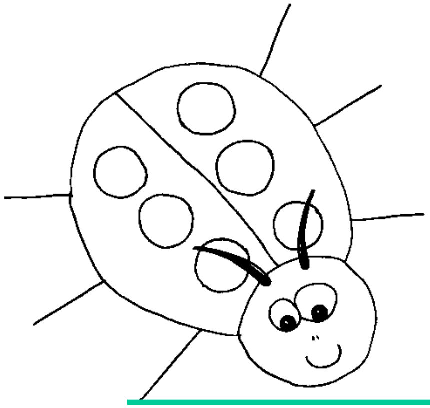
What if all events are low priority?

What happens if a event should be high priority but is given a low priority?

What if events occur out of order?

What if an event does not take place?

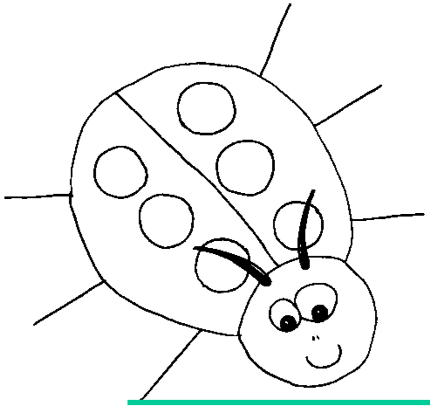
Capture testing ideas



Testing Ideas
Failure Modes

Connector Data access	What if data being polled does not change?
	What if channel to database is closed?
	What if channel to database break during a transaction?
	What if the data access requires a different access authorization or privilege level?
	What if multiple processes or threads share the same data elements and there is a resource contention problem?

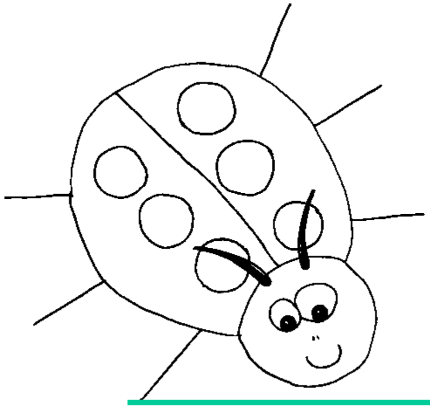
Capture testing ideas



Testing Ideas
Failure Modes

Connector Data access	What if multiple processes or threads share the same data elements and there is a race condition due to the timing or order of operations?
	What if data is locked?
	What if data access does not respond within timeout period?
	What if multiple processes or threads are all waiting for each other to liberate a resource before processing continues leading to a stand-off situation?

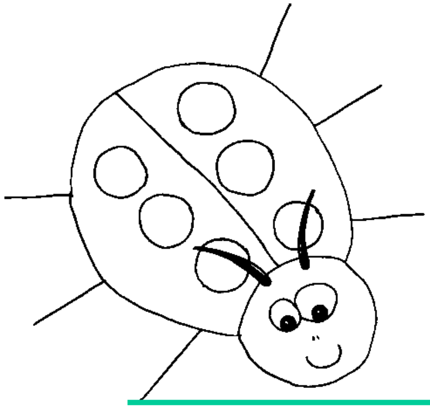
Capture testing ideas



Testing Ideas
Failure Modes

Connector Linkage	What if physical transport layer fails?
	What if connection linkage is noisy and randomly garbles data?
	What if the link it to the wrong place?
	What if the link is shared?
	What if the available bandwidth diminishes?

Capture testing ideas



Testing Ideas
Failure Modes

Connector Stream

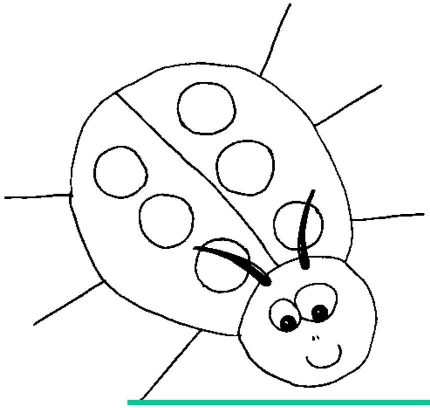
What if streaming process is memory starved?

What if there are too many concurrent threads?

What if the wrong stream is invoked?

What if a stream returns an error code?

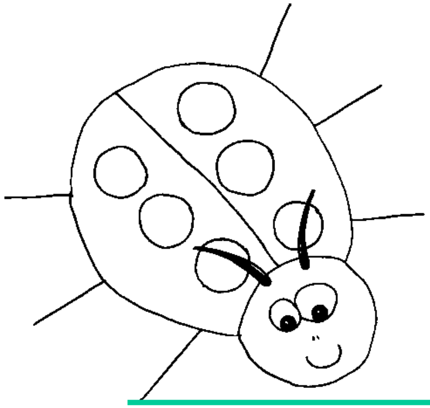
Capture testing ideas



Testing Ideas
Failure Modes

Capture testing ideas

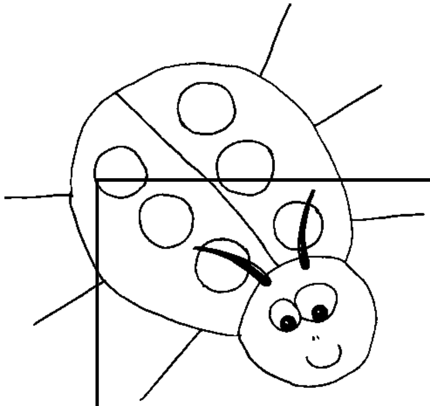
Connector Arbitrator	What if there is a standoff while two processes are waiting for an event?
	What if the priority of the event is misunderstood?
	What if the arbitration process fails blocking access?
	What if the arbitration fails and all processes have concurrent access?
	What if access is never granted because higher priority events continue to win arbitration?



Testing Ideas
Failure Modes

Connector Distributor	What if transaction is routed to the wrong process?
	What if distributor returns error condition to the wrong source?
	Can distributor reroute transaction if one path fails?
	What if multiple routes fail concurrently?

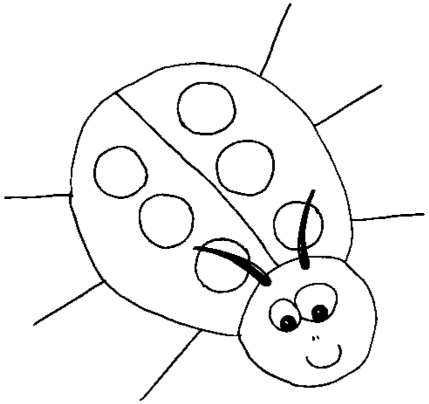
Capture testing ideas



	Adaptability	Accessibility	Auditability	Availability	Continuity	Dependability	Expandability	Functionality	Integrity	Interoperability	Maintainability	Operability	Portability	Reliability	Re-usability	Scalability	Security	Serviceability	Testability	Usability
Application service provider	🟩	🟥	🟥	🟥	🟩	🟨	🟩	🟨	🟨	🟨	🟥	🟩	🟩	🟥	🟩	🟥	🟥	🟨	🟨	🟨
Automatic content generator	🟩	🟨	🟩	🟨	🟩	🟨	🟩	🟨	🟥	🟩	🟨	🟩	🟩	🟥	🟩	🟨	🟩	🟨	🟨	🟩
Customized access	🟩	🟥	🟩	🟥	🟩	🟥	🟩	🟨	🟥	🟩	🟩	🟨	🟩	🟥	🟩	🟨	🟨	🟩	🟩	🟥
Database access	🟨	🟥	🟨	🟥	🟩	🟥	🟨	🟥	🟥	🟨	🟨	🟩	🟨	🟥	🟨	🟥	🟥	🟨	🟨	🟨
Delivery	🟩	🟥	🟨	🟥	🟩	🟥	🟩	🟩	🟥	🟩	🟩	🟩	🟩	🟥	🟩	🟨	🟥	🟩	🟩	🟩
Document access	🟩	🟨	🟩	🟥	🟩	🟥	🟨	🟨	🟥	🟨	🟩	🟨	🟩	🟨	🟩	🟨	🟥	🟨	🟨	🟨
File sharing	🟩	🟨	🟨	🟥	🟩	🟥	🟨	🟨	🟥	🟨	🟩	🟨	🟩	🟩	🟩	🟨	🟥	🟨	🟨	🟥
Informational	🟩	🟨	🟩	🟥	🟩	🟥	🟨	🟨	🟥	🟨	🟨	🟨	🟩	🟥	🟨	🟨	🟨	🟨	🟨	🟨
Interactive	🟥	🟩	🟥	🟨	🟥	🟥	🟩	🟥	🟥	🟥	🟥	🟩	🟥	🟥	🟨	🟨	🟥	🟨	🟨	🟨
Transaction oriented	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟨	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥	🟥
User-provided content	🟩	🟨	🟩	🟨	🟩	🟩	🟨	🟨	🟨	🟨	🟩	🟨	🟩	🟨	🟩	🟨	🟨	🟩	🟩	🟩
Workflow oriented	🟨	🟨	🟥	🟨	🟩	🟨	🟨	🟨	🟥	🟨	🟨	🟨	🟨	🟨	🟩	🟨	🟨	🟨	🟨	🟨

High Focus	🟥
Medium Focus	🟨
Low Focus	🟩

Quality Factors Importance
Different Application Types

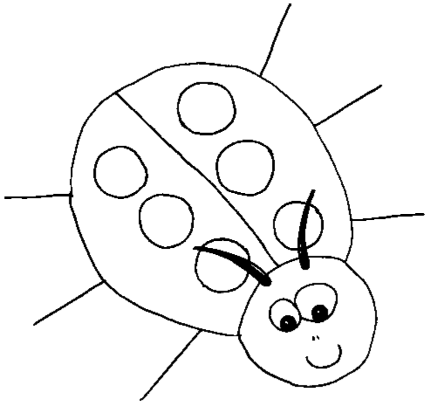


Testing Ideas
Quality Factors

Adaptability and Expandability

- GIST: Confirm that the Wrap-O-Matic can correctly create boxes of chocolates using a competing products manifest. Randomly sample manifests from many competitors.

Capture testing ideas

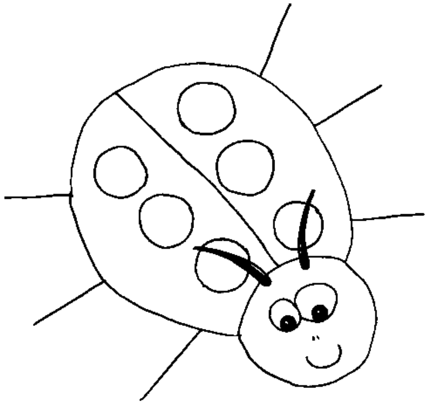


Testing Ideas
Quality Factors

Accessibility

- GIST: Study how operator colour blindness might impact error rates. Explore tasks related to Starting, Stopping, Monitoring Status of or Manifest Entry into the Wrap-O-Matic.

Capture testing ideas

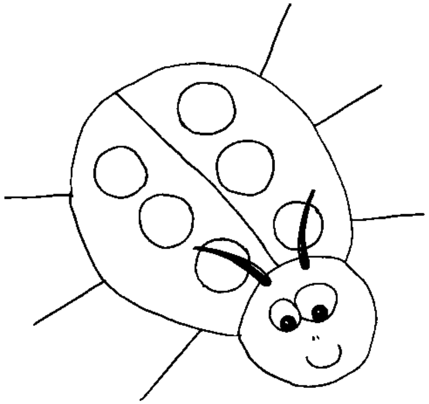


Testing Ideas Quality Factors

Auditability

- GIST: Can an auditor during, or after, a production run, confirm (a) reports match logged data, (b) reports match production results and (c) logged data match production results?

Capture testing ideas

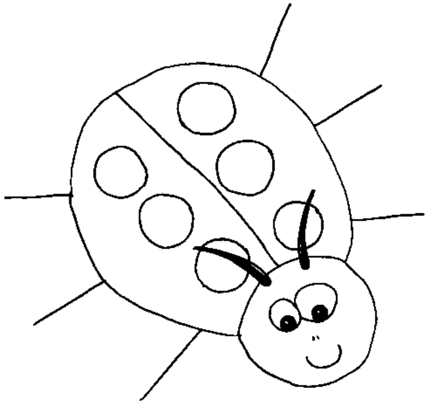


Testing Ideas
Quality Factors

Availability

- GIST: Study Wrap-O-Matic availability by performing several trial production runs at maximum duty cycle. (Availability = Uptime/(Uptime+Downtime)).

Capture testing ideas

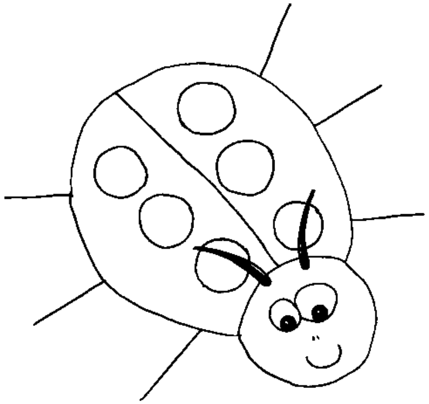


Testing Ideas
Quality Factors

Continuity

- GIST: Confirm that the Wrap-O-Matic can resume operation after pausing for Maintenance, Repairs, Inspections, or Audits. Ensure accuracy of logs, reports, and production results.

Capture testing ideas

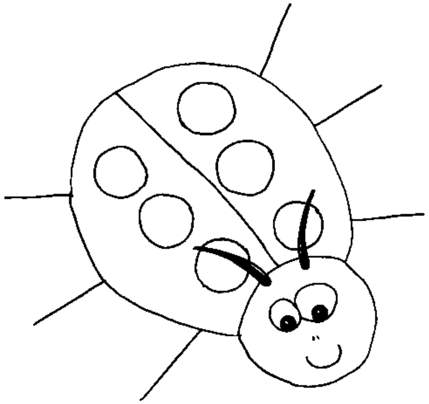


Testing Ideas
Quality Factors

Dependability

- GIST: Confirm that boxes of chocolates are consistently produced correctly for a variety of manifests.

Capture testing ideas

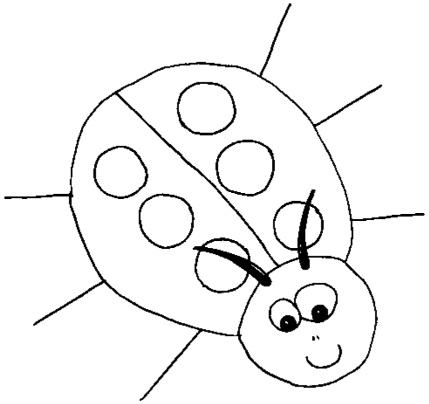


Testing Ideas
Quality Factors

Reliability

- GIST: Study Wrap-O-Matic reliability by performing several trial production runs at maximum duty cycle under harsh, accelerated life conditions. Measure the Mean Time Between Failures.

Capture testing ideas

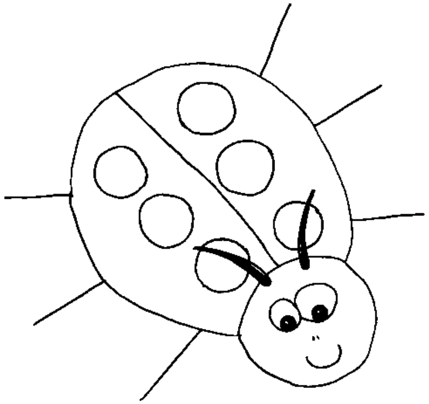


Testing Ideas
Quality Factors

Integrity

- GIST: Explore whether Wrap-O-Matic failures can accidentally corrupt transaction logs or manifests data.

Capture testing ideas

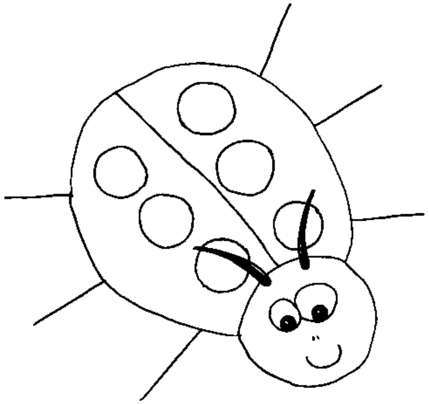


Testing Ideas
Quality Factors

Interoperability

- GIST: Confirm that Wrap-O-Matic reports can be exchanged with and correctly interpreted by the chocolate manufacturer's management information system.

Capture testing ideas



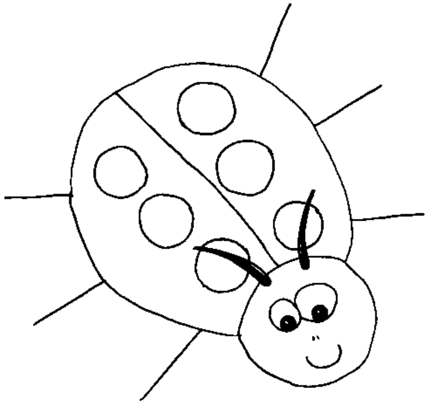
Testing Ideas

Quality Factors

Maintainability and Serviceability

- GIST: Confirm that Wrap-O-Matic firmware can be upgraded to a new revision.

Capture testing ideas

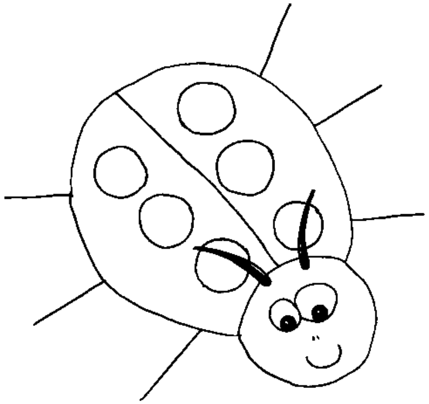


Testing Ideas Quality Factors

Operability

- GIST: Beta test the Wrap-O-Matic at customer sites. Study how well the Wrap-O-Matic works with the customer's manufacturing and business process.

Capture testing ideas

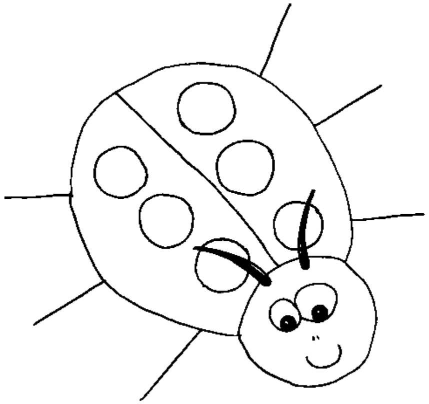


Testing Ideas
Quality Factors

Re-usability

- GIST: Confirm that the Wrapping Engine source code can be reused in alternative hardware environments. Use static analysis techniques to study the source code.

Capture testing ideas

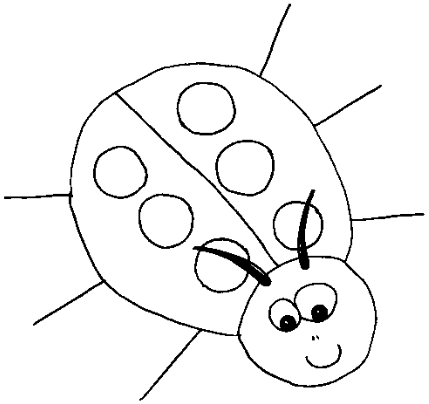


Testing Ideas
Quality Factors

Portability and Compatibility

- GIST: Explore how Wrap-O-Matic operator console software runs on Windows, Mac, and Linux environments.

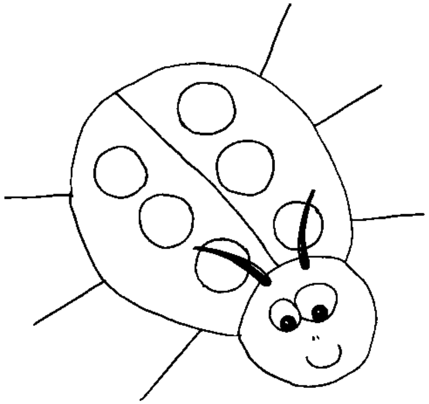
Capture testing ideas



Testing Ideas
Quality Factors

Scalability

- GIST: Experiment with the relationship between the number of wrapping modules and the Wrap-O-Matic throughput. How does the change in throughput relate to the number of Wrapping Modules?

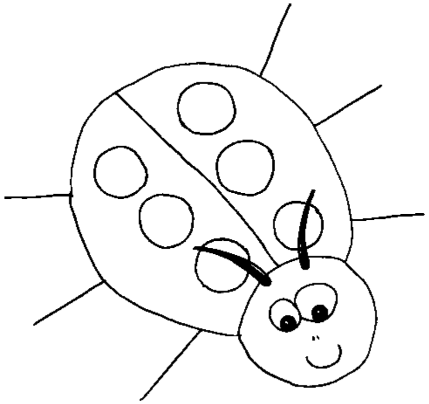


Testing Ideas
Quality Factors

Performance

- GIST: Determine the average throughput (boxes of chocolates per hour) for the most common (Pareto) types of boxes of chocolates.

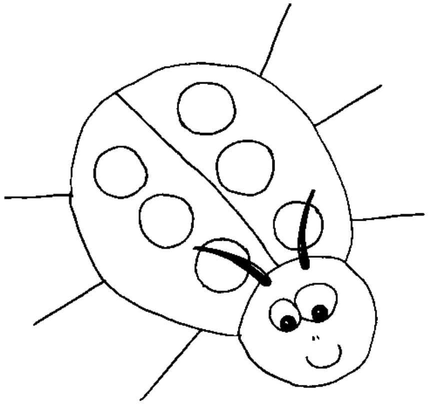
Capture testing ideas



Testing Ideas
Quality Factors

Effectiveness and Efficiency

- GIST: For different production runs: light (no wrapping), typical (normal wrapping), and harsh (complex wrapping) study the Wrap-O-Matic's CPU and memory resource usage over time.

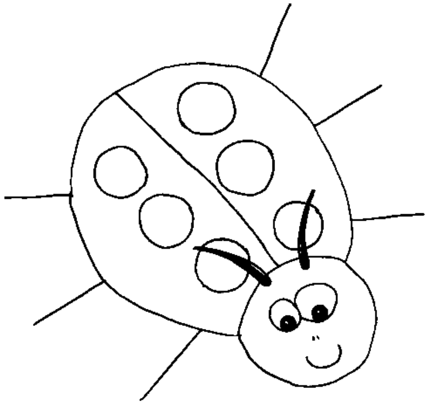


Testing Ideas
Quality Factors

Robustness

- GIST: Assess robustness of Operator Console by user interface attacks and randomized data entry.

Capture testing ideas

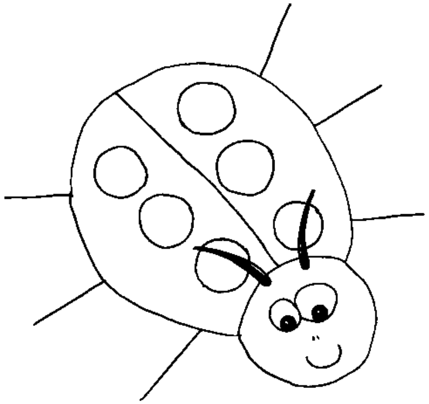


Testing Ideas
Quality Factors

Security

- GIST: Ensure that only authorized users can access and manipulate restricted data. (eg. Loader should not be allowed to enter manifest).

Capture testing ideas

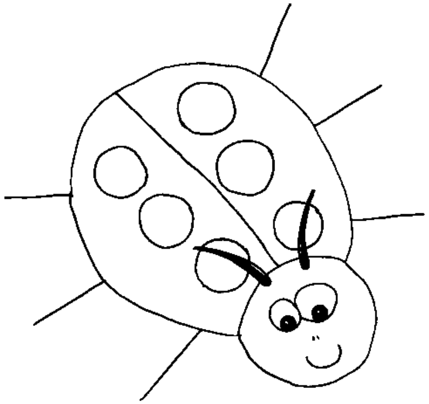


Testing Ideas
Quality Factors

Testability

- GIST: Perform a Wrap-O-Matic design review. Has the software been instrumented to show the progress of each process during a production run?

Capture testing ideas

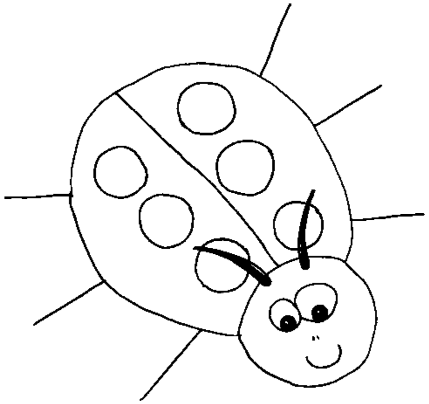


Testing Ideas
Quality Factors

Usability

- GIST: Study how an experienced chocolate factory worker, who has never been trained in using the Wrap-O-Matic, can accomplish the task of entering a manifest with the only guidance coming from on-line help.

Capture testing ideas

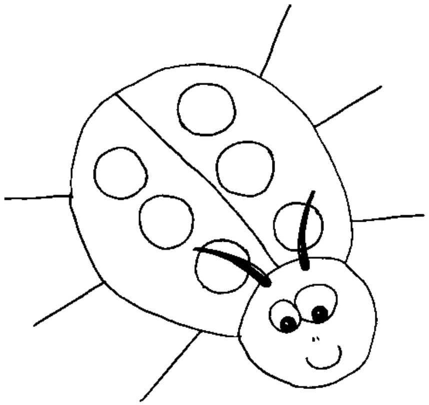


Testing Ideas
Quality Factors

Power Consumption

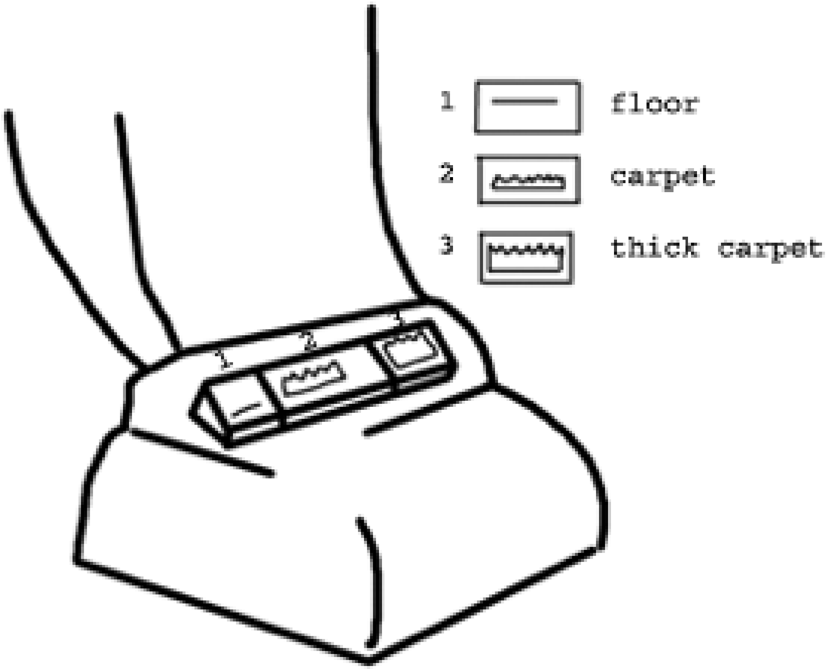
- GIST: Study the battery power consumption by the manifest entry app during entry of typical manifests before a production run.

Capture testing ideas

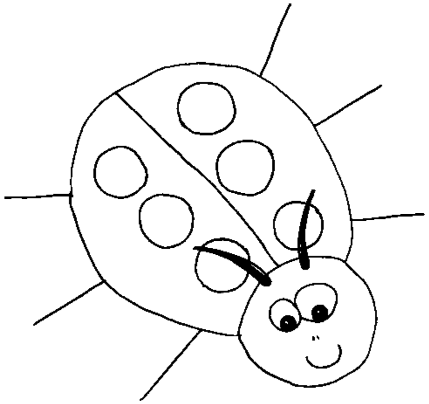


Testing Ideas

- Usage Scenarios
  - Identify classes of users
  - Identify how users will use system
  - Describe scenarios
  - Use Story board or similar approaches
  - Identify variations



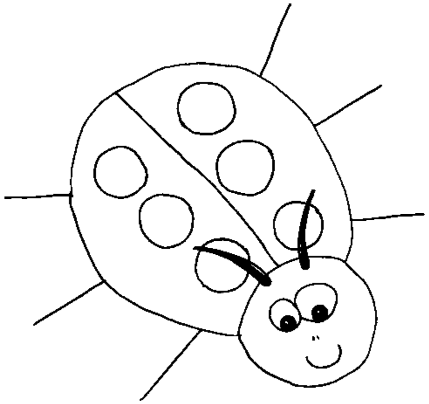
Capture testing ideas



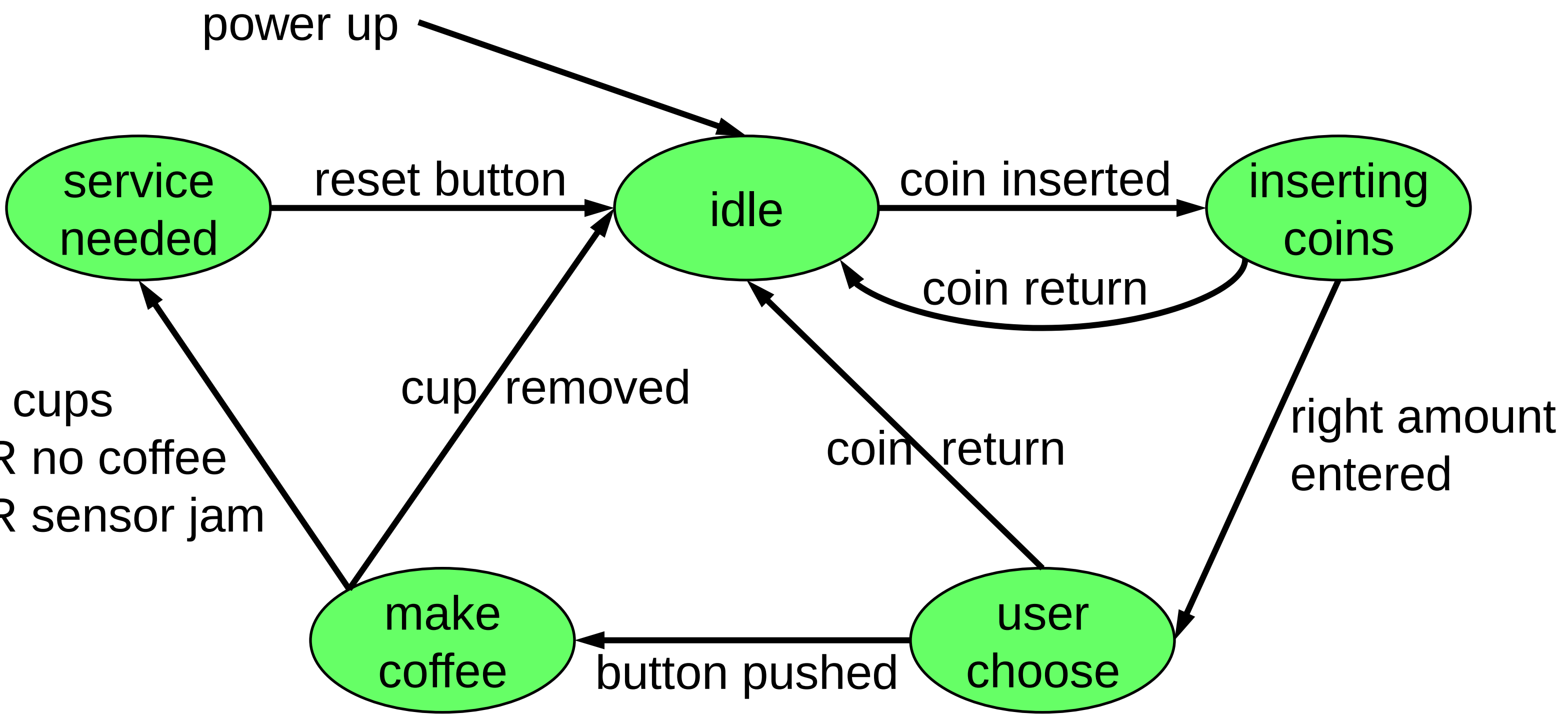
Testing Ideas

- Creative approaches
  - Action verbs
  - Mind Maps
  - Soap Operas
  - Lateral Thinking

Capture testing ideas

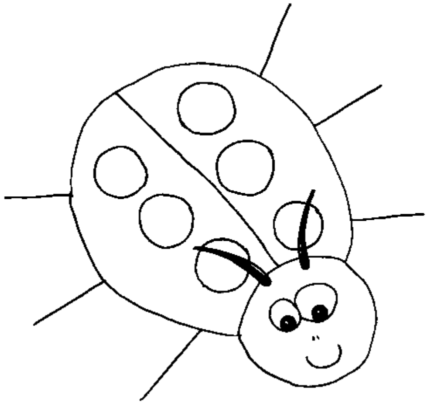


Testing Ideas



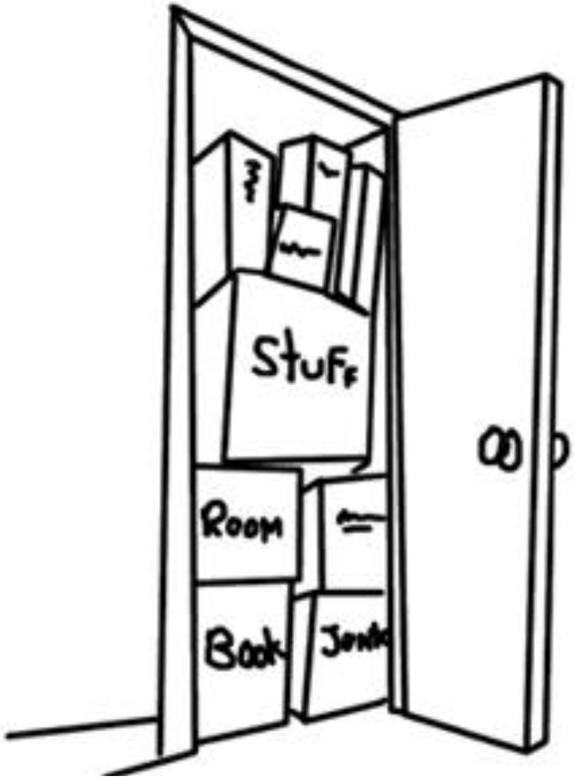
State Models

Capture testing ideas

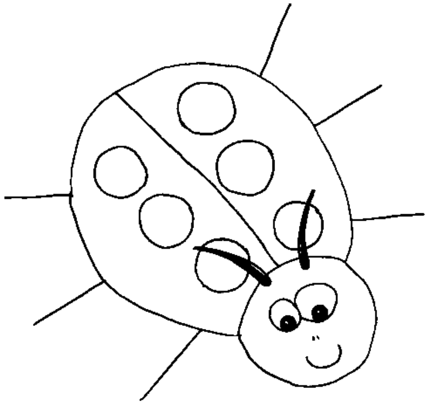


Testing Ideas

- Data
  - Flow
  - Structure
  - Create
  - Update
  - Change

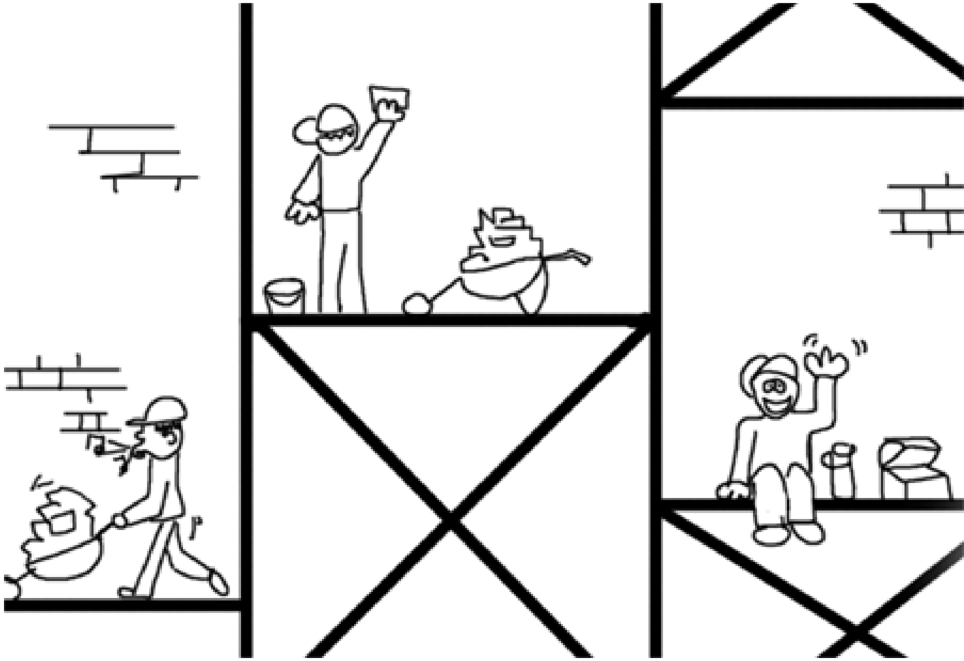


Capture testing ideas

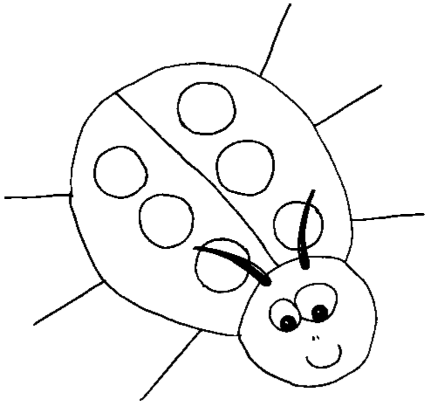


Testing Ideas

- Environment
  - Hardware
  - Software
  - Operating systems
  - Locales
  - Browsers
  - Plug-ins
  - Co-dependent software

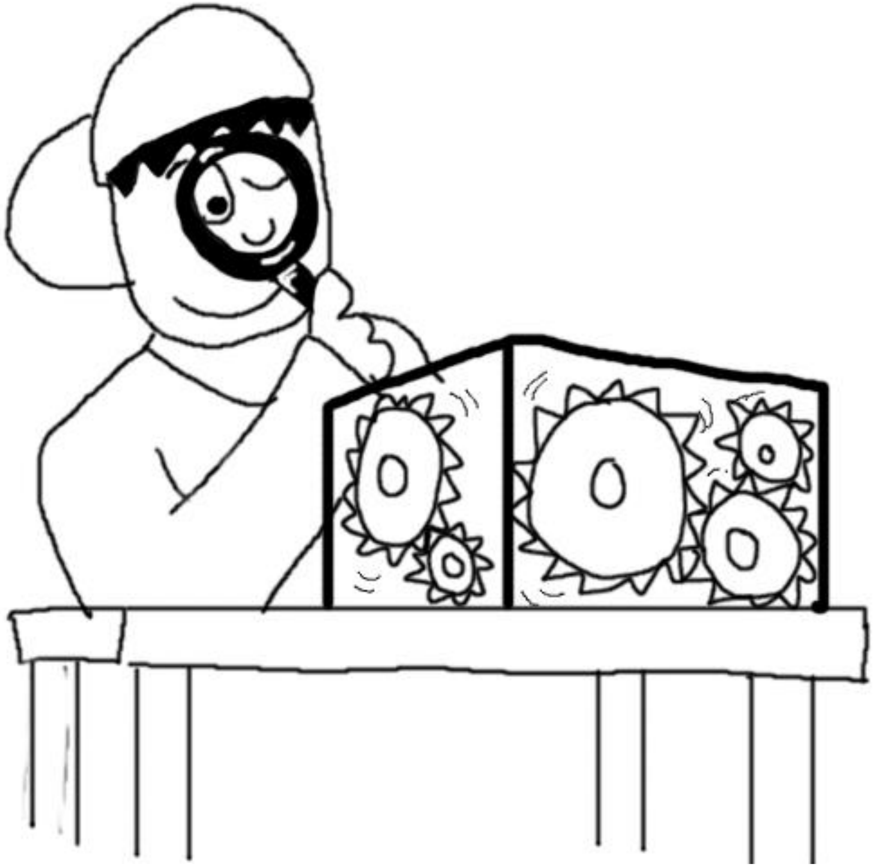


Capture testing ideas

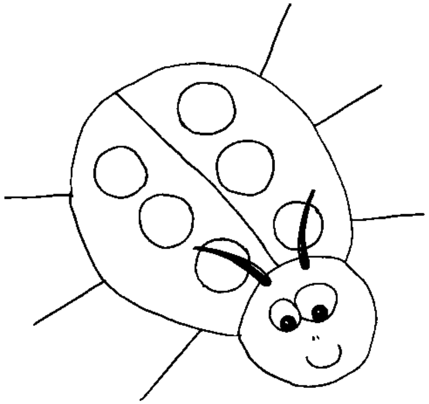


Testing Ideas

- White Box
  - Design
  - Internal structure
  - Code

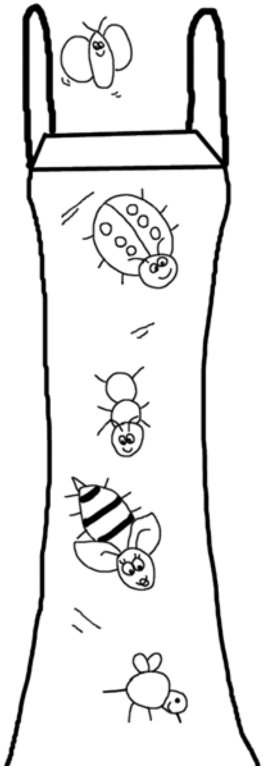


Capture testing ideas

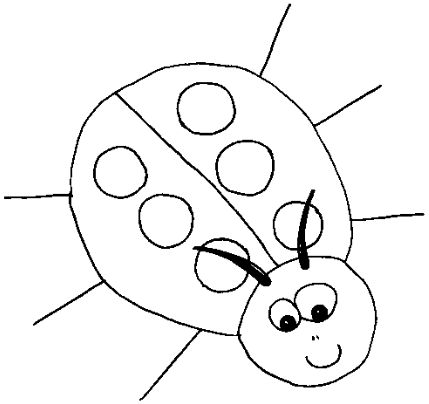


Testing Ideas

- Bug taxonomies
  - Collections of possible bugs
  - Appendix A of *Testing Computer Software, Kaner, Falk, Nguyen*
  - Boris Biezer Taxonomy Otto Vinter manages
  - Shopping cart taxonomy Giri Vijayaraghavan



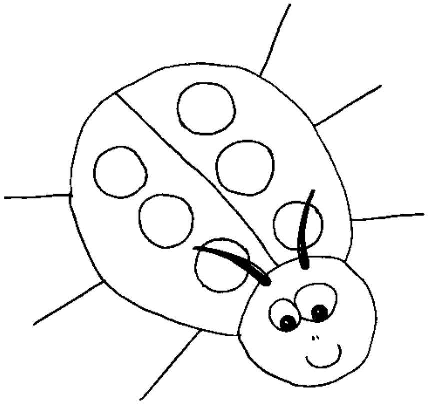
Capture testing ideas



Testing Ideas

- Across Story Relationships
  - Interference
  - Resource sharing
  - Inconsistent behaviour
  - Out of order

Capture testing ideas

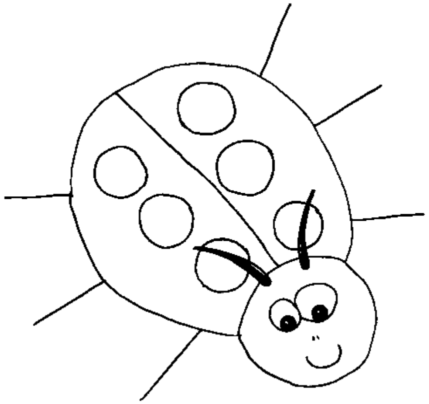


Testing Ideas

- Software Breaking
  - James Whitaker, How to Break Software
  - Create a fault model
  - Identify weakness
  - Apply attack
  - User interface attack
  - System interface attack
  - Data layer attack
  - Security attack

Capture testing ideas

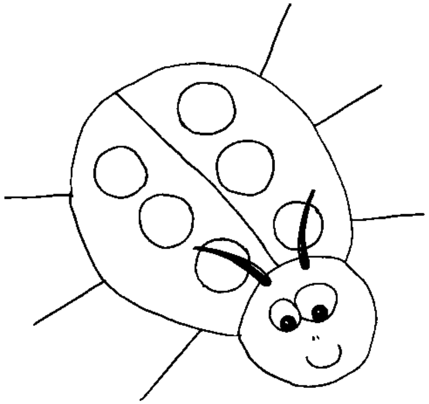
© 2020 Robert Sabourin



Testing Ideas

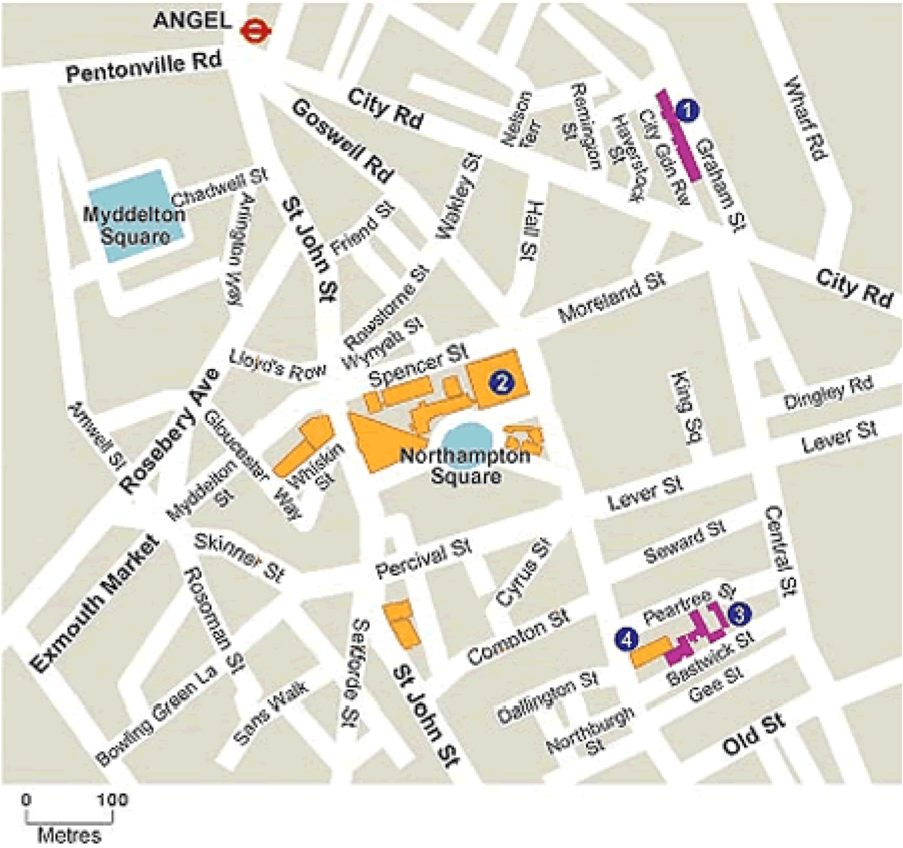
- End to End Testing
  - Exercise entire process chain to complete a transaction
  - Automatic steps
  - Manual steps
  - External systems
  - Third-party systems

Capture testing ideas

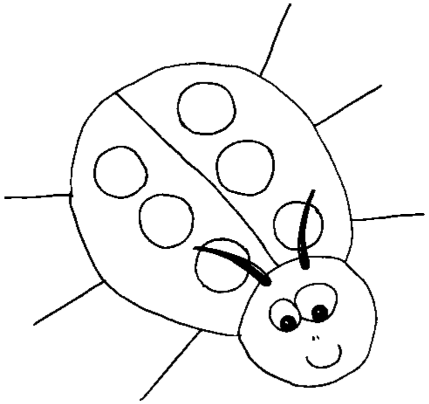


Testing Ideas

- Sequences
  - Explore paths
  - Vary
  - Operation order
  - Sequences
  - Valid
  - Invalid
  - Multiple
  - Concurrent

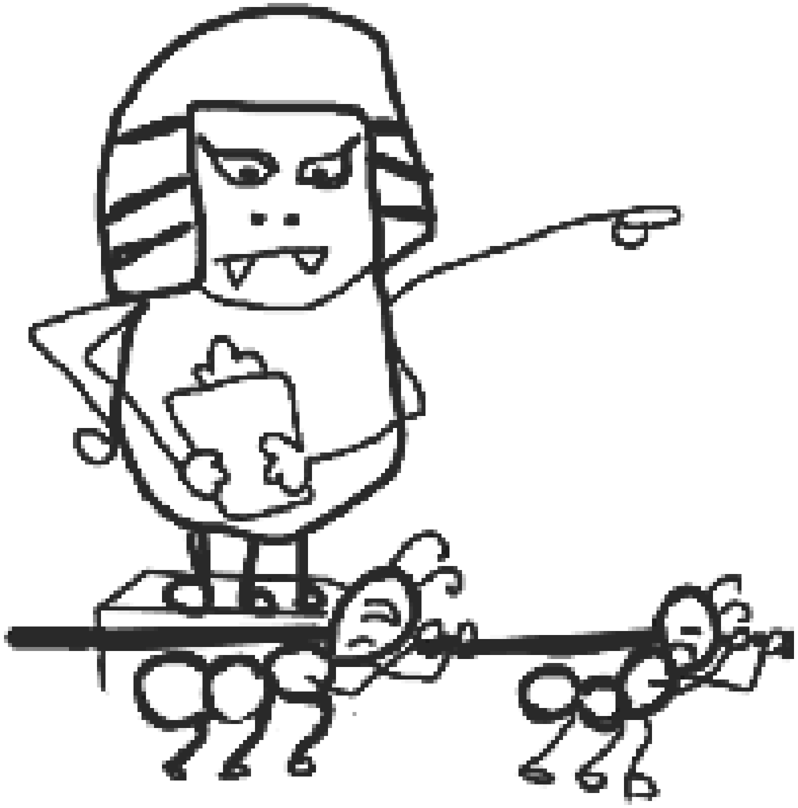


Capture testing ideas



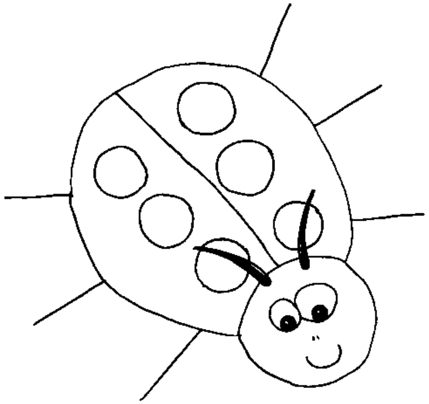
Testing Ideas

- Business Rules
  - Decisions
  - Limits
  - Constraints
  - Process Models
  - Transaction logic
  - Getting things done
  - Value



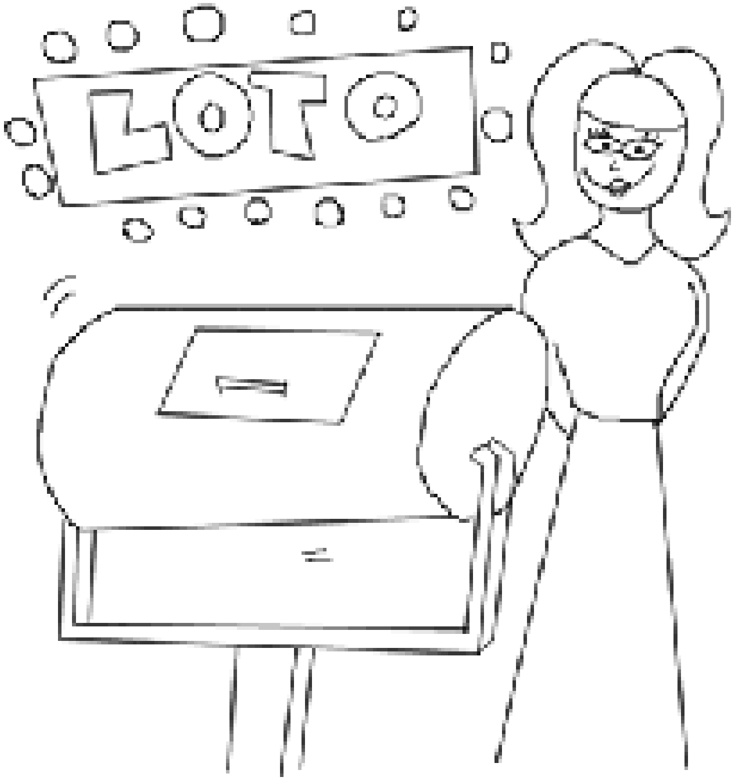
Capture testing ideas

© 2020 Robert Sabourin

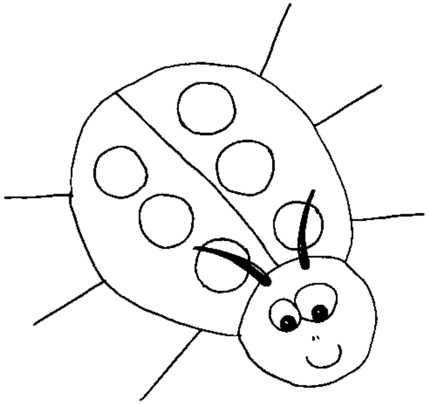


Testing Ideas

- Combinations
  - Multiple variables
  - Selections
  - Options
  - Configurations
  - Permutations
  - Pareto
  - Pairwise

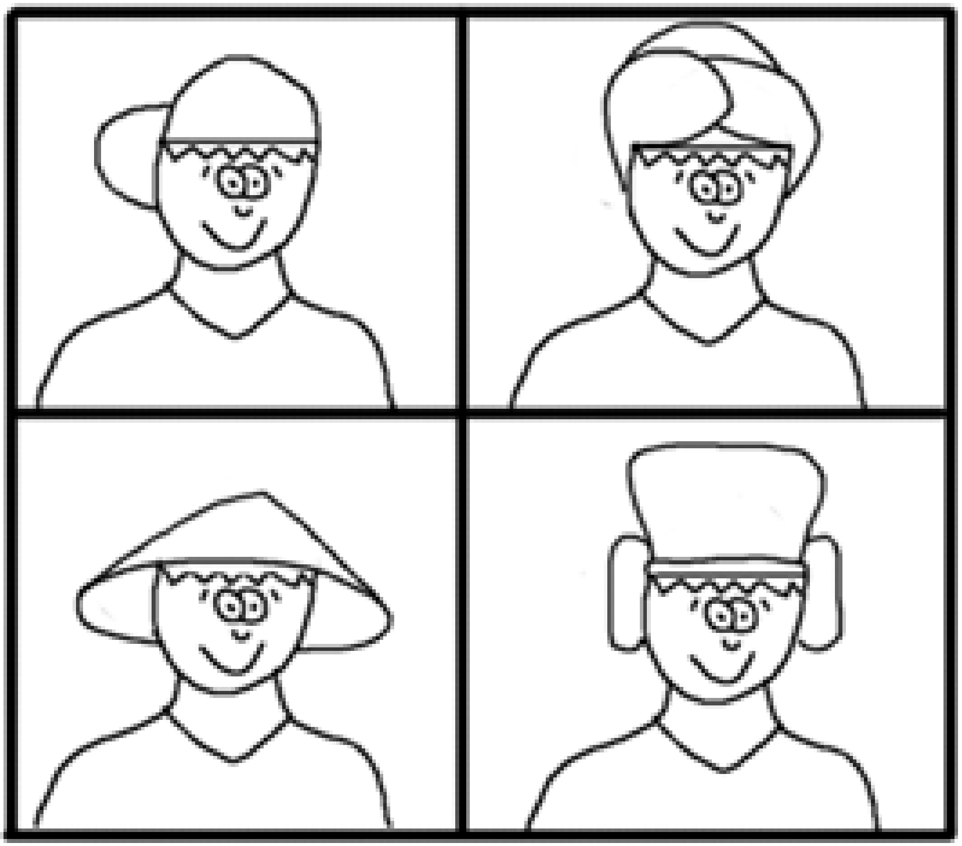


Capture testing ideas

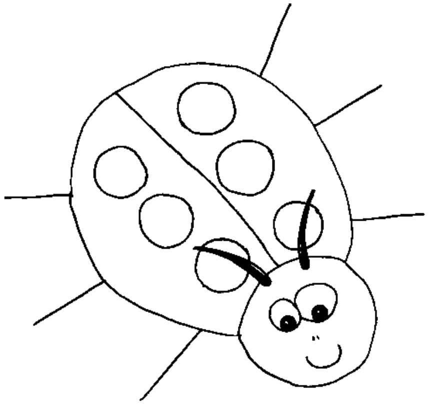


Testing Ideas

- Internationalization & Localization
  - Language
  - Culture
  - GUI
  - Locale Specific Rules
  - Collation
  - Searching
  - Sorting
  - Test coding

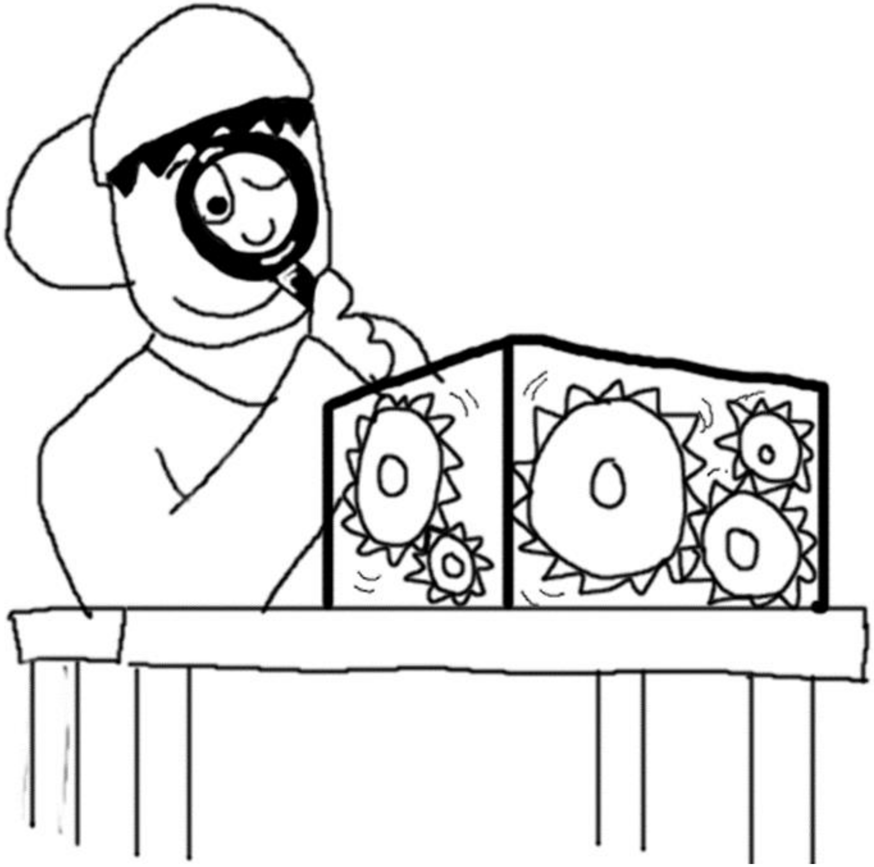


Capture testing ideas

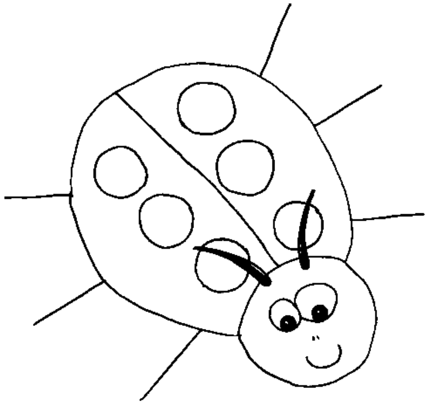


Testing Ideas

- Unit Test
  - Code
  - Structure
  - Data
  - Coverage
  - Design



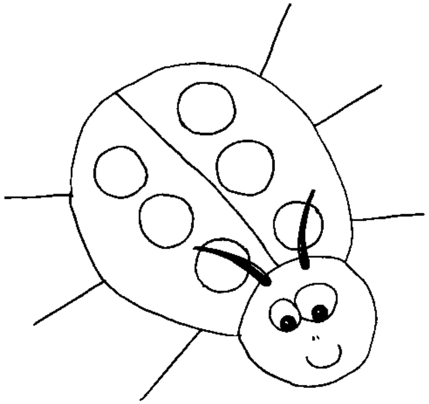
Capture testing ideas



Testing Ideas

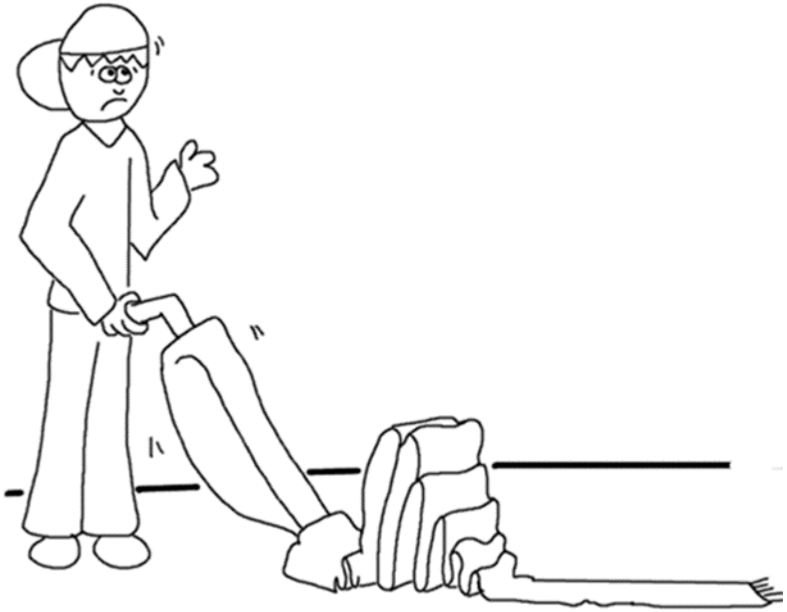
- Test Oracles
  - Truth
  - Assess correctness
  - Requirements
  - State machines
  - Subject matter experts
  - Designs
  - Domain experts
  - Heuristics
  - Validation

Capture testing ideas

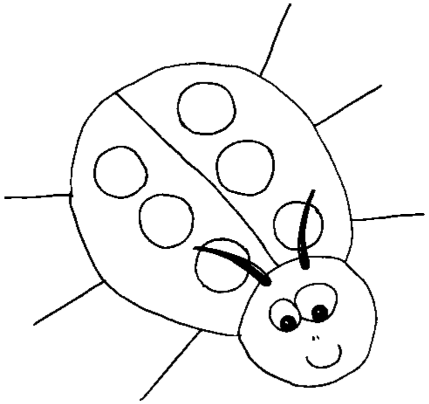


Testing Ideas

- Boundary Tests
  - Value ranges
  - Edge conditions
  - Extremes
  - Point behaviours change
  - Limits in time
  - Security

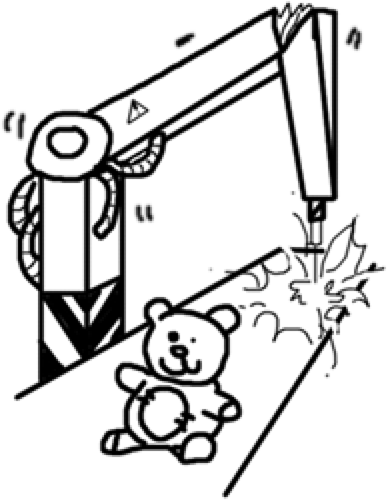


Capture testing ideas

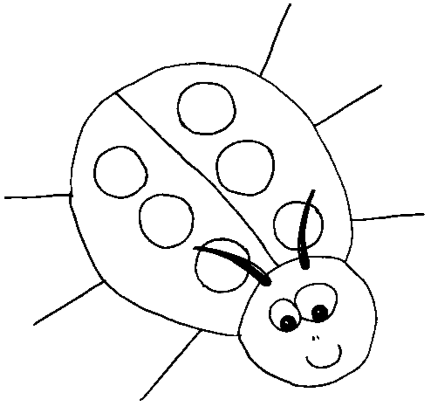


Testing Ideas

- Automation
  - Having tools to automate part of software testing can suggest test ideas
  - Repetitive
  - Hard to observe outcomes
  - Complex computations
  - Do a lot of transactions
  - Create and compare data

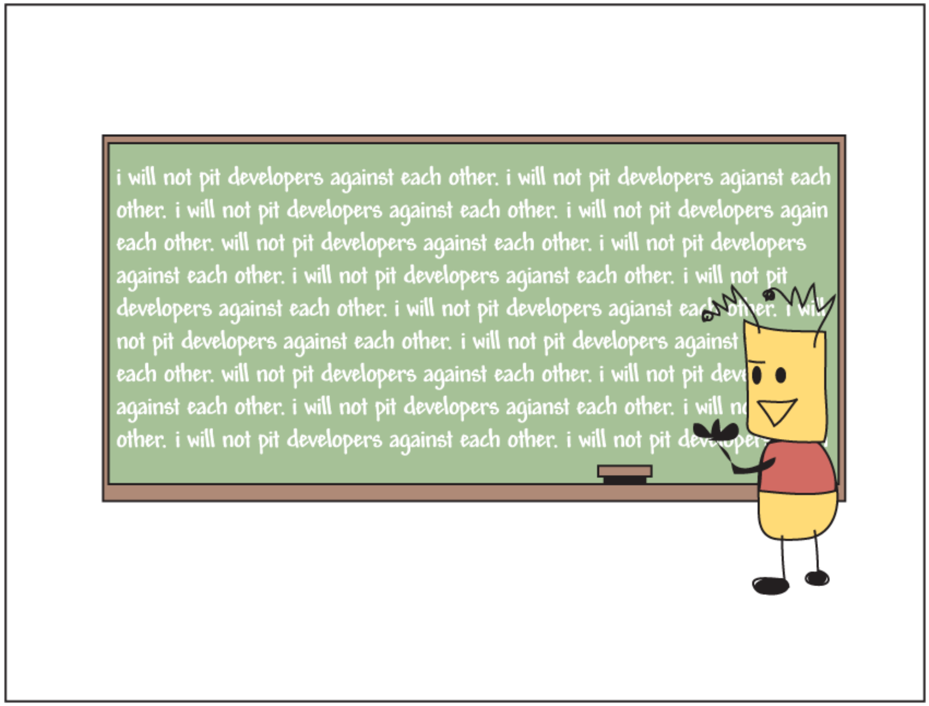


Capture testing ideas



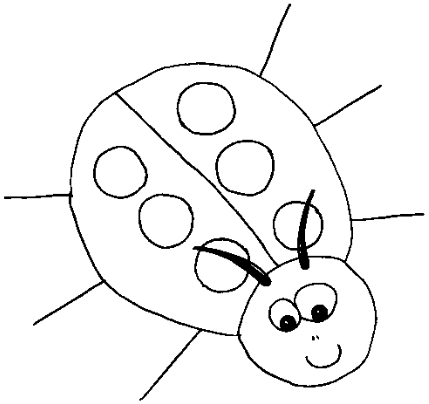
Testing Ideas

- Regression
  - Continuous Integration
  - Smoke Tests
  - Release Readiness
  - Story Tests
  - Unit Tests
  - Observe & Control


i will not pit developers against each other. i will not pit developers agianst each
other. i will not pit developers against each other. i will not pit developers again
each other. will not pit developers against each other. i will not pit developers
against each other. i will not pit developers agianst each other. i will not pit
developers against each other. i will not pit developers agianst eac other. i wi
not pit developers against each other. i will not pit developers against
each other. will not pit developers against each other. i will not pit deve
against each other. i will not pit developers agianst each other. i will n
other. i will not pit developers against each other. i will not pit developer

Capture testing ideas

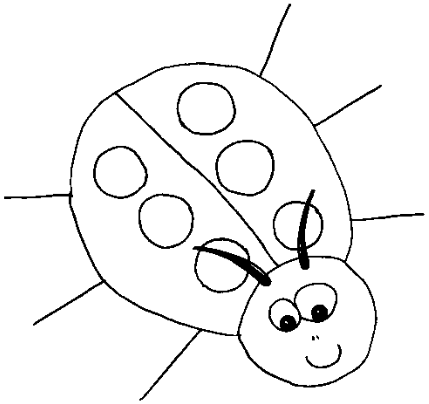
© 2020 Robert Sabourin



Triage

- Criticality
- Resources
- Trade offs
- Credibility

Triage testing ideas



Which test?

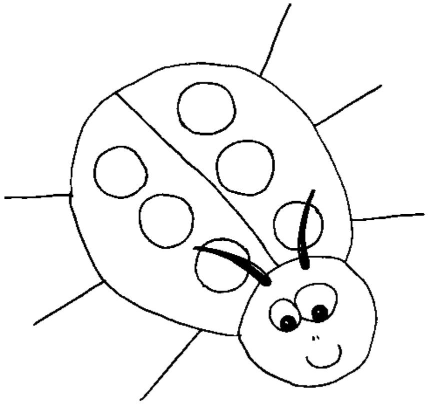
- *Impact estimation*
  - –For each test idea guesstimate:
    - benefit of implementation
    - consequence of implementation
    - benefit for not implementing
    - consequence of not implementing
  - –How credible is the information?

Triage testing ideas

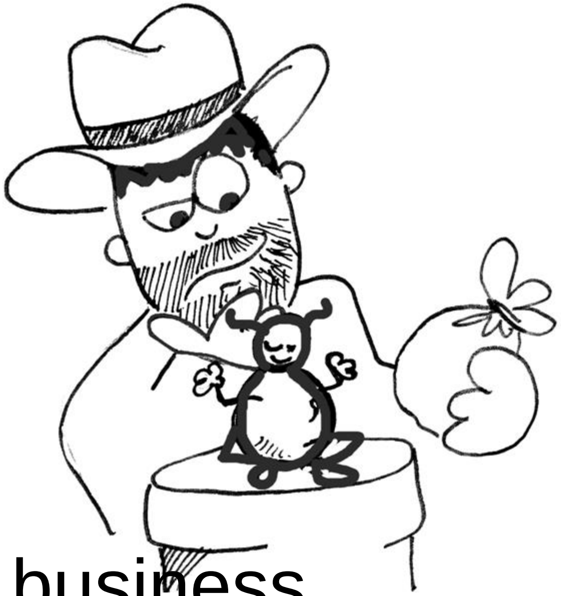
Understanding Complex Technology Quantitatively By Tom Gilb

How to Decide?

Rank	Credibility
0.0	Wild guess, no credibility
0.1	We know it has been done somewhere
0.2	We have one measurement somewhere
0.3	There are several measurements in the estimated range
0.4	The measurements are relevant to our case
0.5	The method of measurement is considered reliable
0.6	We have used the method in-house
0.7	We have reliable measurements in-house
0.8	Reliable in-house measurements correlate to independent external measurements
0.9	We have used the idea on this project and measured it
1.0	Perfect credibility, we have rock solid, contract- guaranteed, long-term, credible experience with this idea on this project and, the results are unlikely to disappear



Which test?

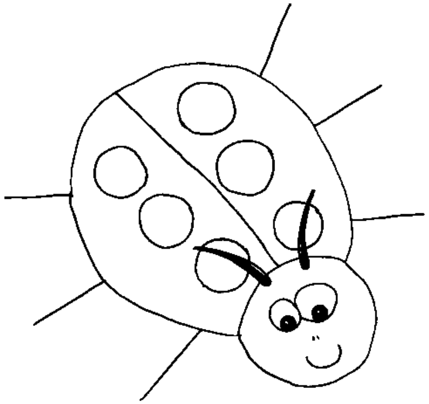


Test Idea Rejection – What If?

–If the cost/benefit does not make business sense then consider implementing:

- *part of the test*, could that lead to part of the benefit at a more reasonable cost?
- *more than the stated test*, would that generate more benefit?
- a different test than the stated idea, could that generate more benefit for less cost?

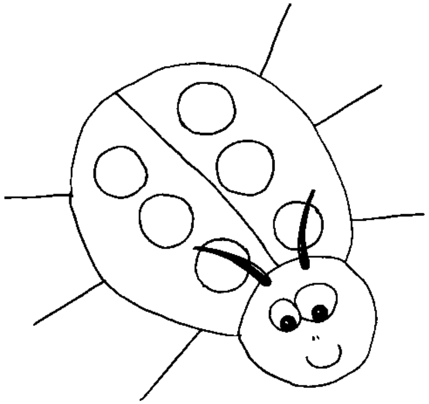
Triage testing ideas



Test Triage

- Test Triage
  - – Turbulent Projects
    - High Frequency
    - Daily +++
  - – Agile Projects
    - On demand
    - At stand up meeting
  - – Stable Products
    - Periodically
    - Same as “bug review”

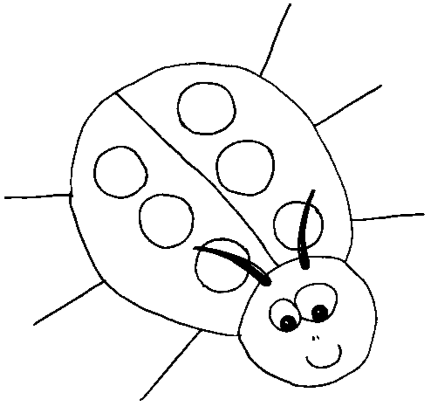
Triage testing ideas



Test Triage

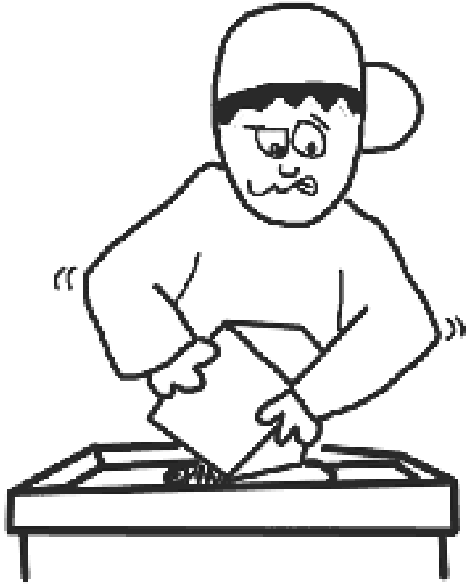
- Review
  - – New Context
  - – New Info
  - – Bugs
  - – New testing ideas

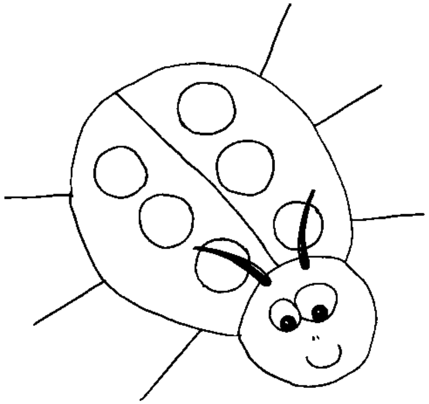
Triage testing ideas



Test Triage

- Review
  - – New Context
    - Business
    - Technical
    - Organizational
    - Cultural

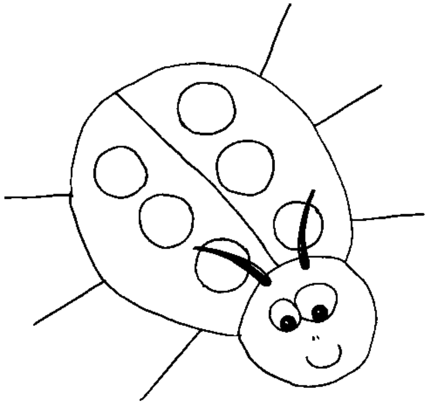




Test Triage

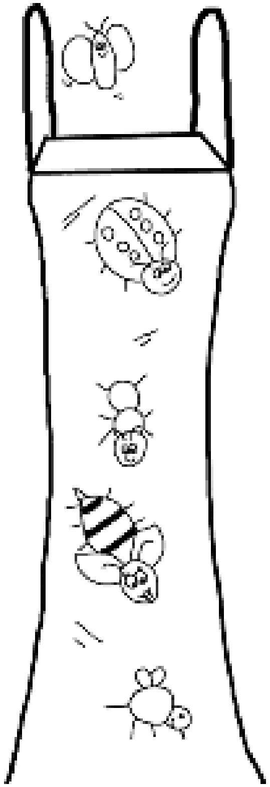
- Review
  - – New Info
    - Test Findings
    - Development
    - Other

Triage testing ideas

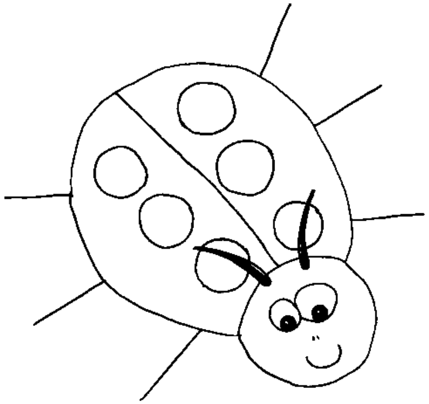


Test Triage

- Review
  - Bugs
    - New
    - Fixed
    - Causes
    - Patterns

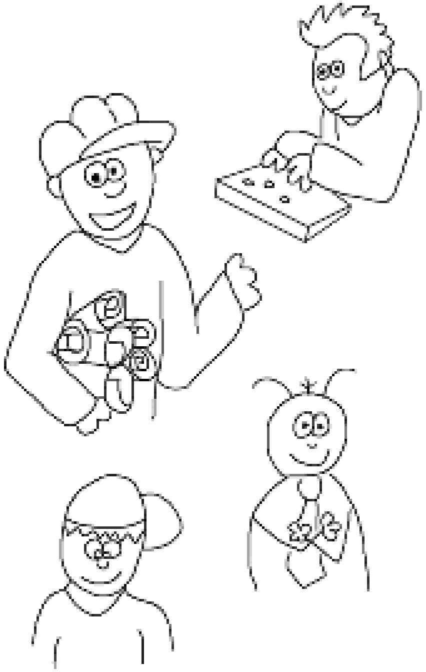


Triage testing ideas

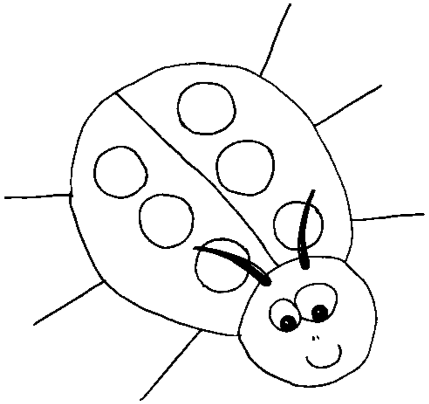


Test Triage

- Review
  - New testing ideas
    - Testers
    - Developers
    - Support
    - Trainers
    - Administrators
    - Customers
    - Users

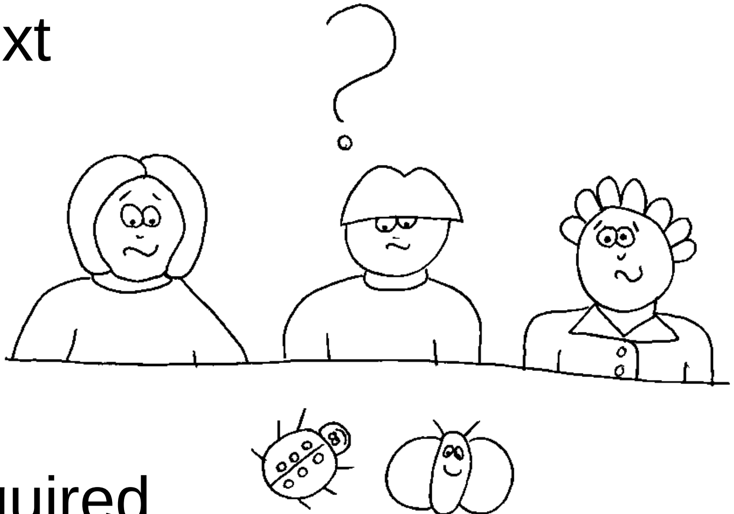


Triage testing ideas

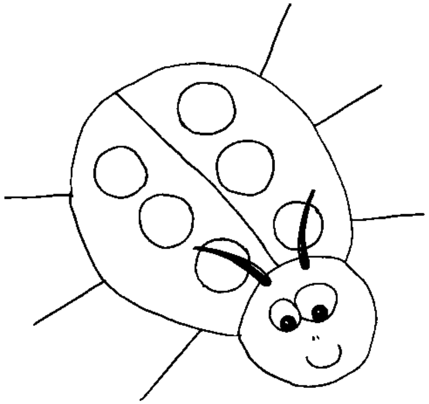


Test Triage

- Allocate Testing Assignments to Testers
  - Make sure testers know context
  - Best thing to test
  - Best person to test it
  - Best people to explore it
  - Best lead
  - Are subject matter experts required



Triage testing ideas

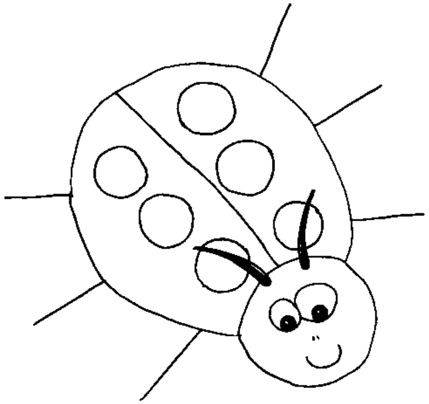


Test Triage

Life of a test idea

a. Comes into existence
b. Clarified
c. Prioritized
   a. Test Now (before further testing)
   b. Test before shipping
   c. Nice to have
   d. May be of interest in some future releas
   e. Not of interest in current form
   f. Will never be of interest
d. Integrate into a testing objective

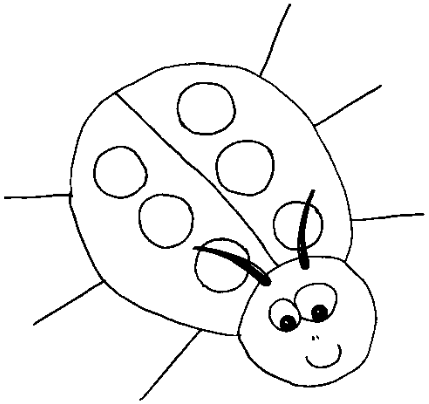
Triage testing ideas



Deciding what not to test?

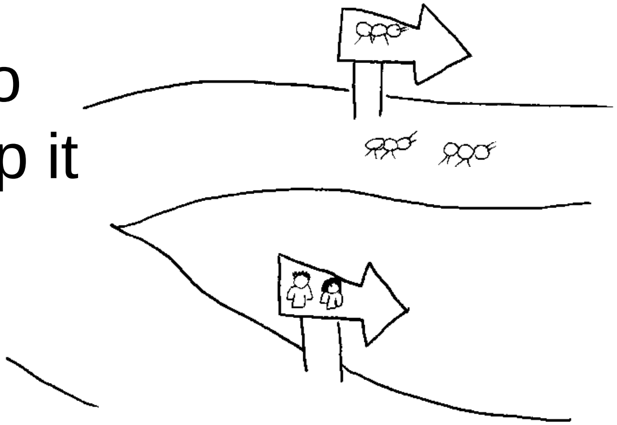
- Time pressure
  - Should we skip a test?
  - If test failed could system still be of value to some stakeholder?
  - If test was skipped could important bugs have been otherwise found?

Triage testing ideas

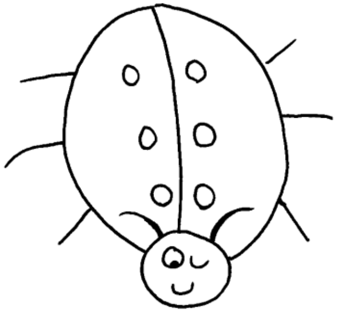


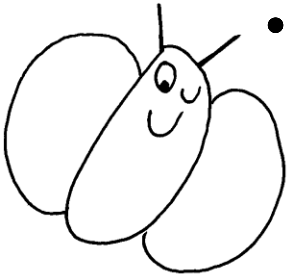
Bottom Line

- *My experience* is that it is better to <u>omit a test on purpose</u> than to skip it because you ran out of time or forgot about it!



Get Started Right





- Systematically collecting, evaluating and triaging testing ideas helps me decide <u>what not to test - *at least for now*</u>?