Effective Test Strategies

Anton Semenchenko

Anton Semenchenko

Anton Semenchenko

Founder of communities
www.**COMAQA**.BY,
www.**CoreHard**.by,
www.**InterIT**.by,
www.**ITUp**.by;
co-founder of company
www.**DPI.Solutions, CSO**;
«tricky» manager at **EPAM Systems.**
15+ years of experience in IT, main specialization:
Automation, C++ and lower development,
management, sales.

Community's audience

Testing specialists (manual and automated)

Automation tools developers

Managers and sales specialists in IT

IT-specialists, thinking about migrating to automation

Students looking for perspective profession.

Community goals

Create unified space for effective communication for all IT-specialists in the context of automated testing.

Your profit

Ability to listen to reports from leading IT-specialists and share your experience.

Take part in «promo»-versions of top IT-conferences in CIS for free.

Meet regularly, at different forums, community «offices», social networks and messengers.

info@comaqa.by

https://www.facebook.com/comaqa.by/

http://vk.com/comaqaby

+375 33 33 46 120
+375 44 74 00 385

www.CoreHard.by

Community's audience

«Harsh» C++ developers & co, IoT, BigData, High Load, Parallel Computing

Automation tools developers

Managers and sales specialists in IT

Students looking for perspective profession.

Community goals

Create unified space for effective communication for all IT-specialists in the context of «harsh» development.

Your profit

Ability to listen to reports from leading IT-specialists and share your experience.

Take part in «promo»-versions of top IT-conferences in CIS for free.

Meet regularly, at different forums, community «offices», social networks and messengers.

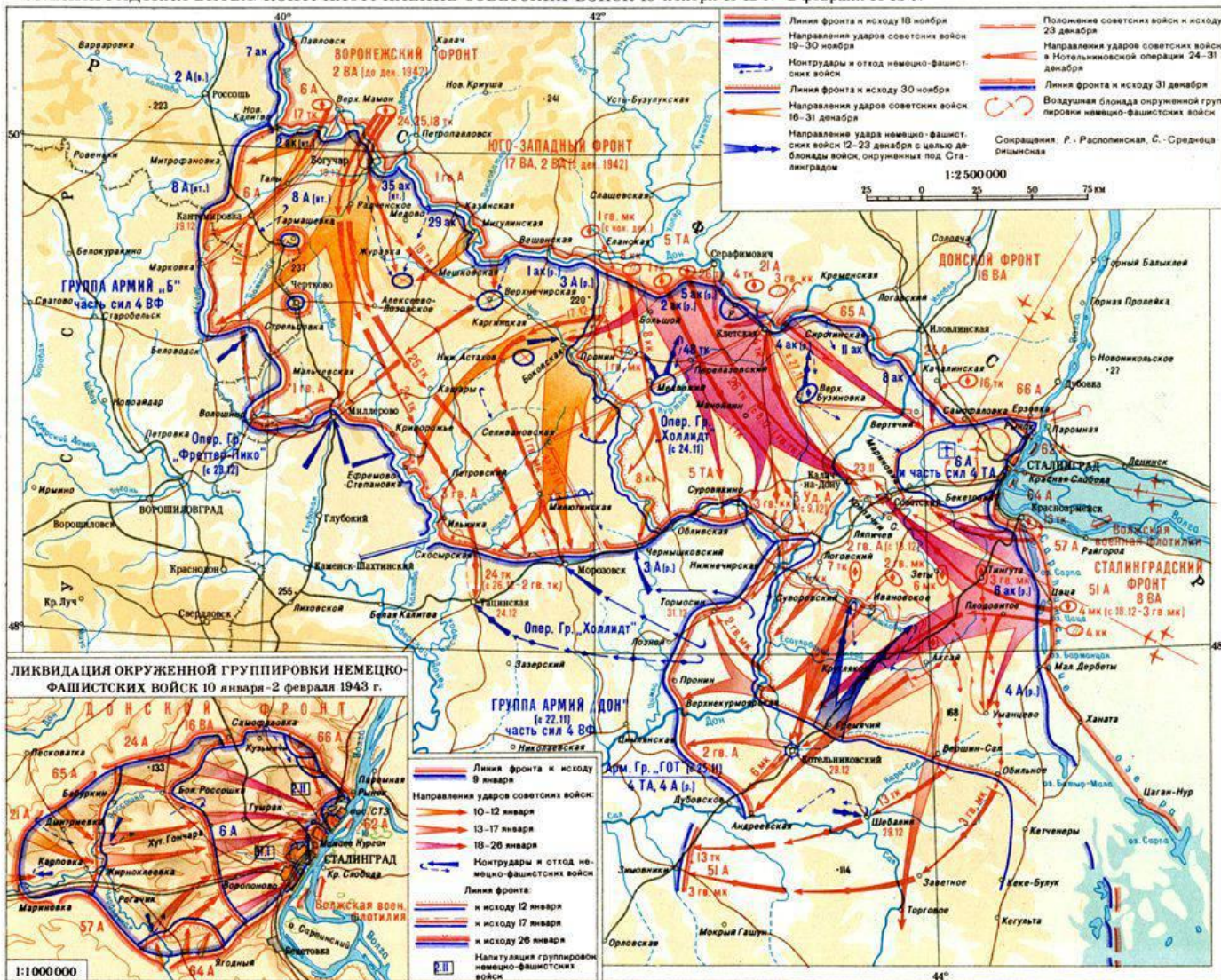info@corehard.by
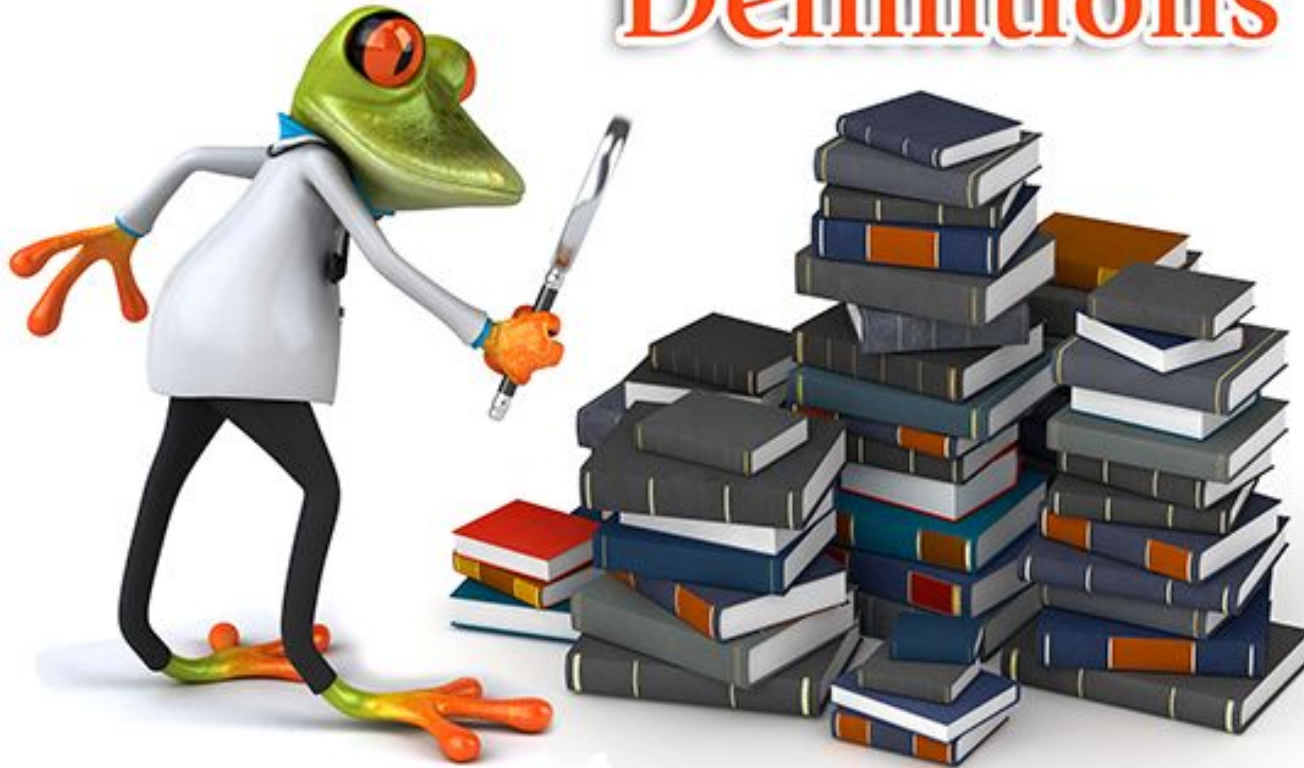
https://www.facebook.com/corehard.by/

+375 33 33 46 120
+375 44 74 00 385

Strategy!

Сталинградская битва. Контрнаступление советских войск 19 ноября 1942 г. – 2 февраля 1943 г.

Definitions: right or wrong?

1. A Test Strategy document is a high level document and normally developed by project manager
2. The Test Strategy document is a static document meaning that it is not updated too often
3. Test strategy is a set of guidelines that explains test design and determines how testing needs to be done
4. Some companies include the "Test Approach" or "Strategy" inside the Test Plan, it is usually the case for small projects. For larger projects, there is one Test Strategy document and different number of Test Plans for each phase or level of testing.

Definitions: right

3.   Test strategy is a set of guidelines that explains test design and determines how testing needs to be done

4.   Some companies include the "Test Approach" or "Strategy" inside the Test Plan, it is usually the case for small projects. For larger projects, there is one Test Strategy document and different number of Test Plans for each phase or level of testing.

RIGHT!

Goals of our conversation

Learn how to

- analyze **needs of customer**
- **elaborate effective** test **strategy**
- **analyze the situation** on current project, **using metrics**
- define **realistic improvements** based on performed analysis
- **persuade customer** in improvements implementation

What's next:

on your current projects

- elaborate **improvements**
- **implement** them
- **measure** the real value of them

Measure value of improvement

Compare particular **metric**
> **before** improvement
> and
> **after**

WHY DO I PERSONALLY NEED THAT?

BECAUSE

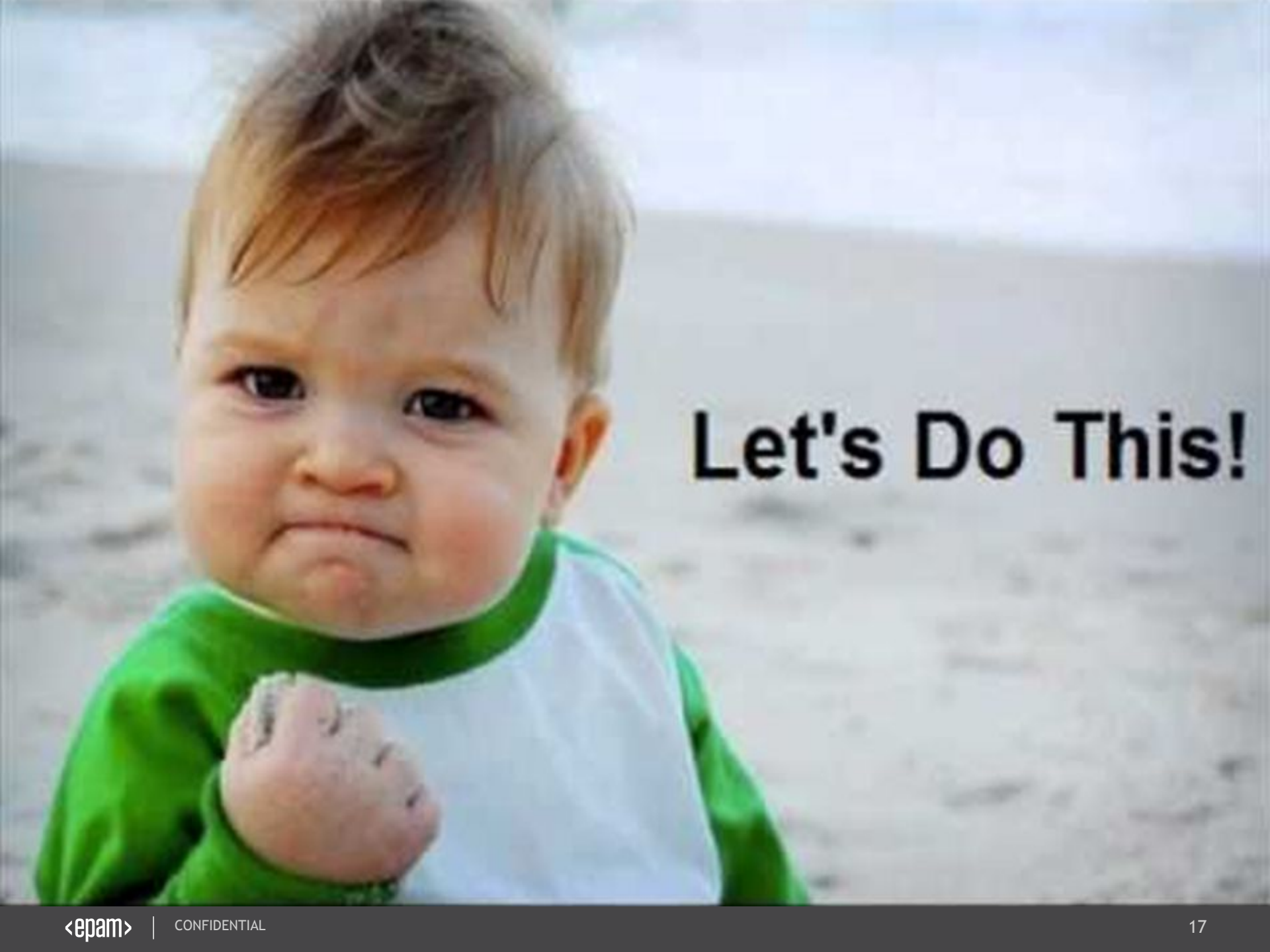1. Professional **self development** (learn something new)
2. **Career development**
3. **Solve problems** on real projects
4. **Prestige**: the most successful projects do it
5. **Get confirmation** of good testing quality in numbers
6. **Visibility, recognition**
7. **Workload optimization**
8. Project **controllability** increase
9. **Relationship** built with all affected stakeholders

PROJECT EXAMPLES

- http://www.ipipeline.com/
- https://www.supplyon.com/en/
- http://www.trizetto.com/

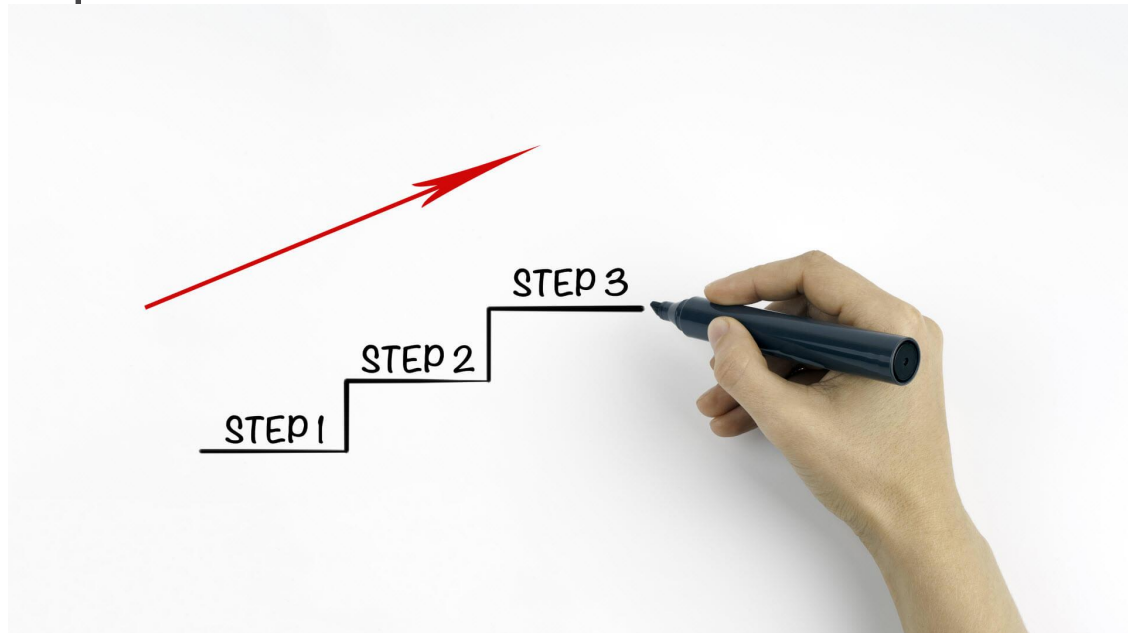Let's Do This!

PRE-STEPS

1. **Testing mission**
   1. The reasons for testing
   2. Is the answer to question "Why do we test"
   3. By people – who are they

2. Context Analyses
   1. Finding out what's important

3. Test Strategy
   1. What and how

TESTING MISSION DEFINITION

- The reasons for testing

- Is the answer to question "Why do we test"

- By people – who are they

TESTING MISSION EXAMPLES

- If you don't know what value testing brings, it's very difficult to do good testing

- Testing missing is the answer to the question: "**Why** do **we test**"

- Standard \ good examples:
  - Contribute by finding important problems (focus of efforts, prioritization)
  - Provide quality related information

- Bad example:
  - Test team is just responsible for testing the product \ We just have to do this

DIFFERENT TESTING MISSIONS

- Because we have clear business goals, time \ budget limitations and have to prioritize our efforts
  - Contribute by finding important problems

- Because business \ we need data for decision support
  - Provide quality related information

- Because we have to:
 - Test team is just responsible for testing the product

THE "SO" TRICK

- When you have a vague mission, like "Test the product"

- Then add "so" and explanation
  - So they can be addressed to get happier customers and fewer support calls

- Rephrase once more
  - So we can find important (from business point of view) problems

- Then you are closing in on a meaningful missing, where stakeholders can add more information:
  - So we can take a well-informed decision
  - So Product Risks have been explored, so we don't get unpleasant surprises

ANALYSIS OF CUSTOMER NEEDS

What is important for customer

Time

Money

Visibility
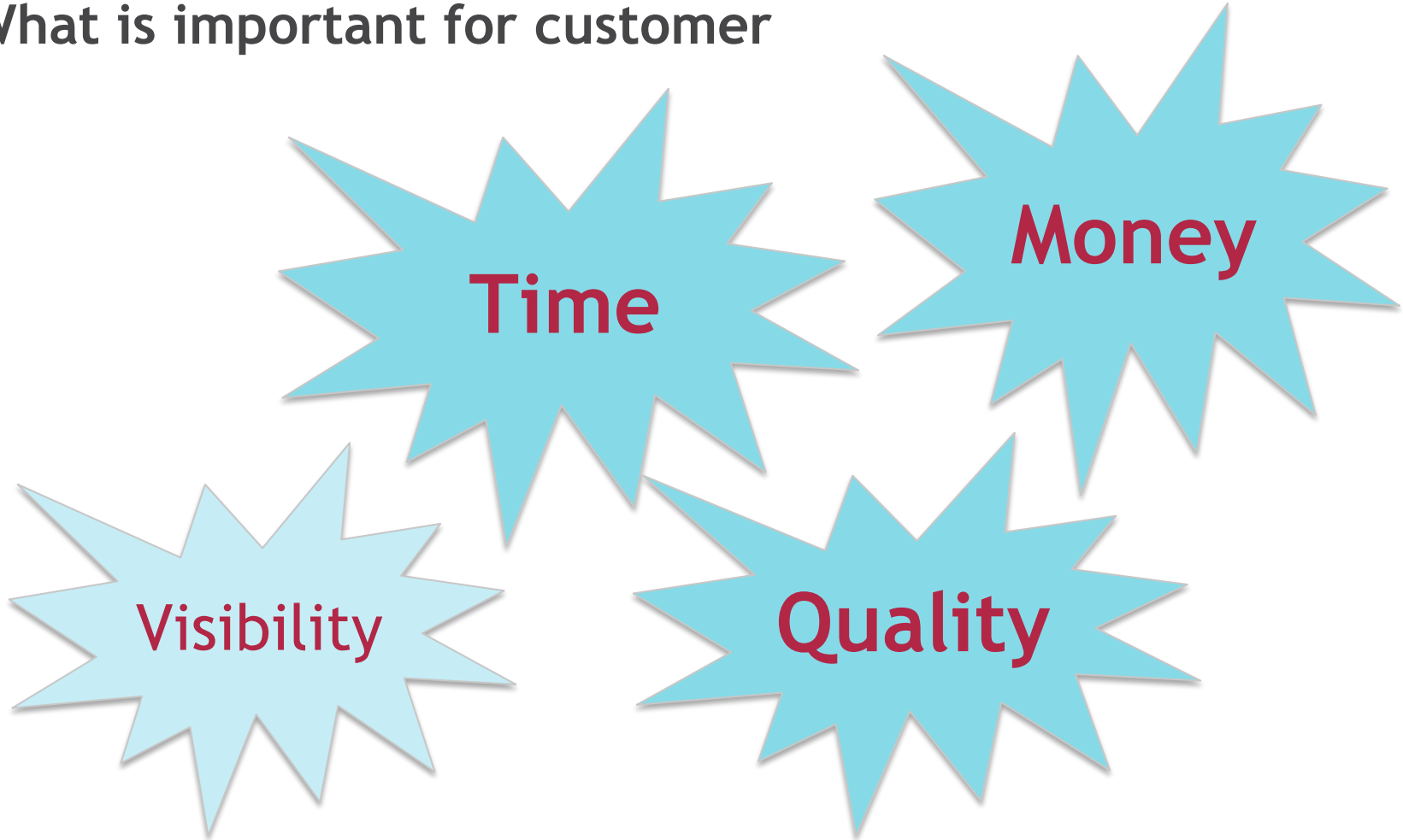
Quality

Based on customer priorities Test Lead creates **Test Strategy**

WHAT IS IMPORTANT AND WHY?

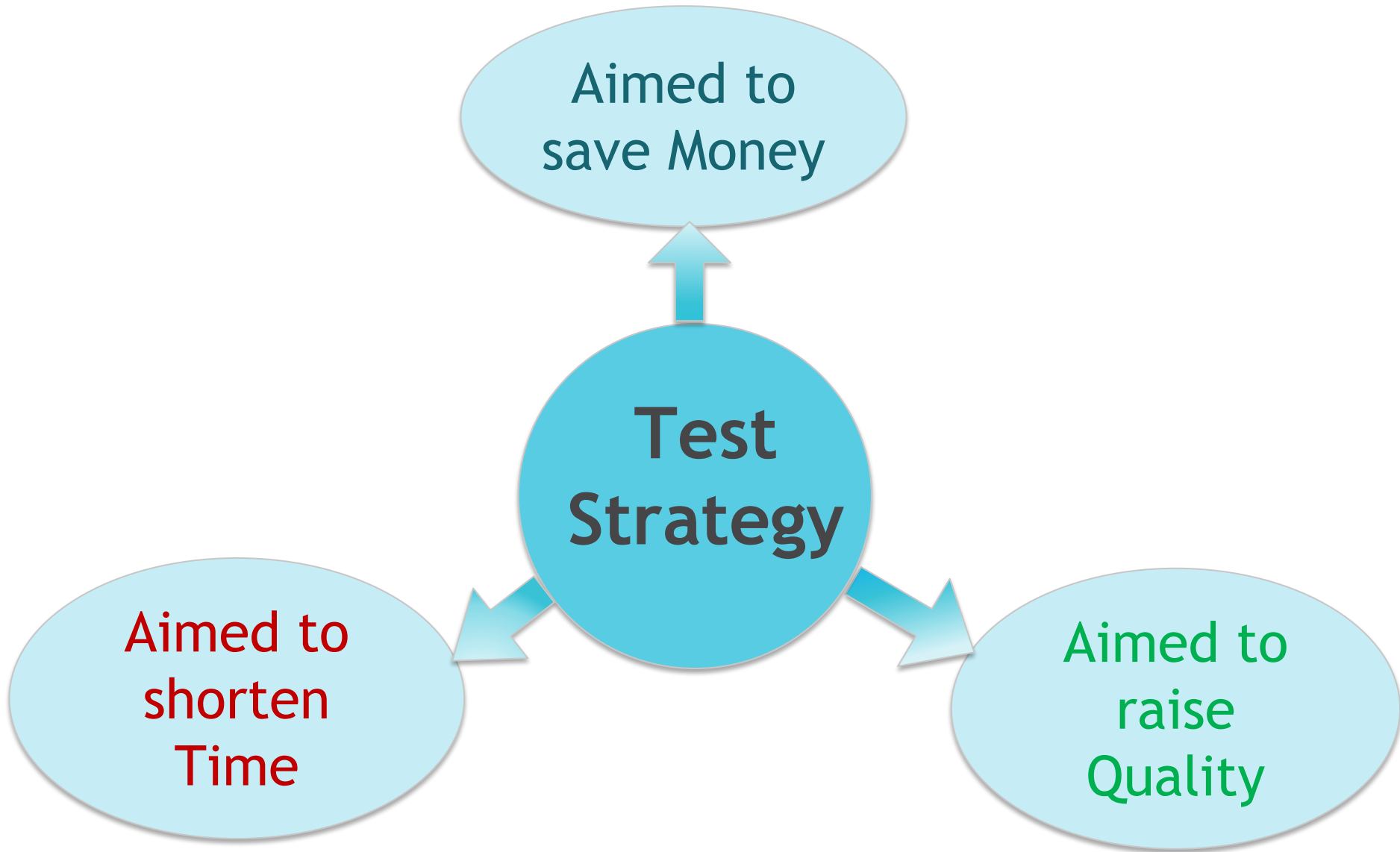1. **Always:**
   - Time
   - Money
   - Quality

2. **Sometimes**
   - Visibility

   For example Support Projects - because **visibility raises customer trust!**

WHICH TEST STRATEGY IS BETTER?

Aimed to save Money

Test Strategy

Aimed to shorten Time

Aimed to raise Quality

STANDARD TEST STRATEGIES EXAMPLES

Different customers care about different parameters!
Something is much more critical.
The priority is important!

Examples:
- **Time**: E-Commerce projects
- **Money / Scope**: Start-Up projects with limited budget
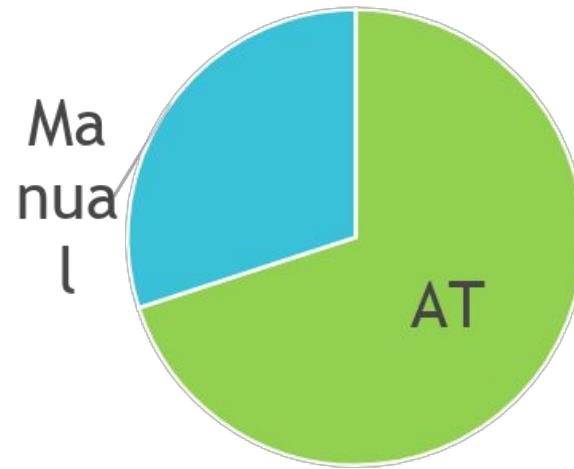- **Quality**: Financial, Healthcare, Embedded, Military

WHICH TEST STRATEGY IS BETTER?

Product "X"
Test Strategy 1

- 1000 TCs
- 20% of them is AT

Product "X"
Test Strategy 2

- 1500 TCs
- 70% of them is AT

THERE IS NO RIGHT ANSWER 😊

There is no right answer!

Again, what matters for customer:
- What is cheaper? (**money**) **1**
- Where tests will be executed faster? (**time**) **2**
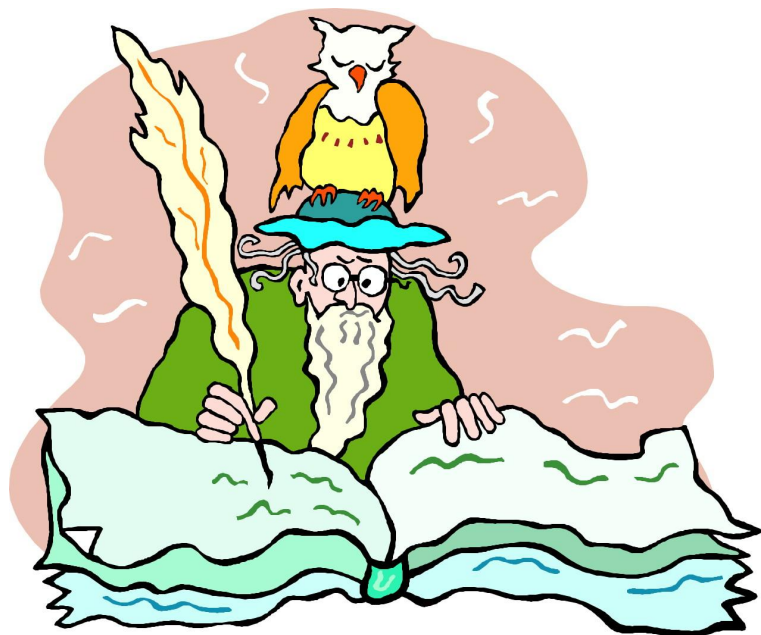- Where more functionality is covered? (**quality**) **2**

Automation has several main purposes:
- Save money
- Speed up release
- Human factor minimization
- + Additional Automation benefits

First of all, Test Strategy is *not necessarily a document.*
It is an ongoing process, that initiates your *though process* and helps to discover possible risks and plan how to mitigate them.

VS

"TYPICAL" TEST STRATEGY

- **Test types**: New Feature, Functional, Regression, etc;
- **Levels**: Smoke, Critical Pathes, Extended;
- Acceptance criteria: no Criticals and Blockers;
- **Process**: Planning > Design > Execution > Reporting

- **This strategy is not effective in 90% of cases, though valid in 100% of cases.**

- **A string of test technique buzzwords is not a test strategy!**

WHAT IS AN EFFECTIVE TEST STRATEGY

Effective Test Strategy helps team to **define:**
- goals
- risks
- test coverage
- approaches
- testing processes
- entry and exit criteria
- tools usage
- environment
- KPIs
- metrics
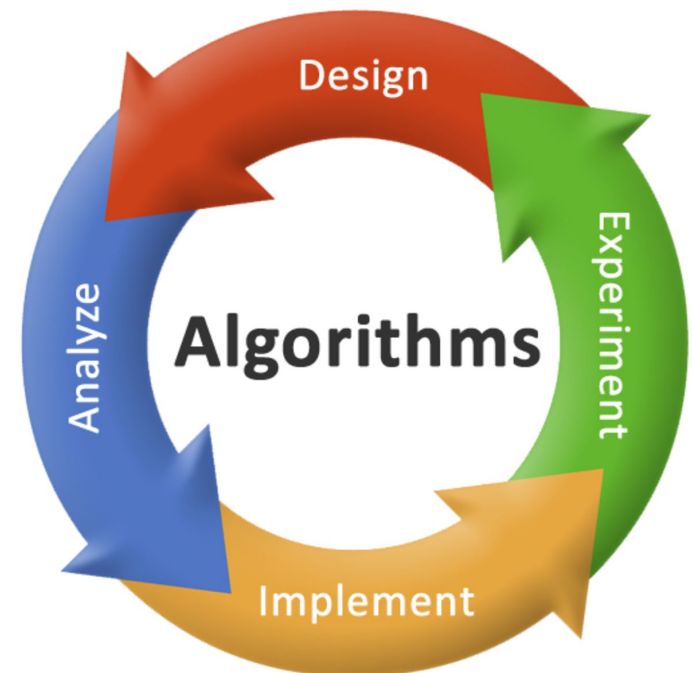
WHAT IS AN EFFECTIVE TEST STRATEGY

Good Test Strategy answers the primary **strategic questions**:

1. **How will you ensure and assess product quality?**

2. **How does test team mitigate particular project risks and address particular project goals?**

HOW TO CREATE EFFECTIVE TEST STRATEGY

1. Learn about the product and **goals**
2. Analyze **scope** and **timelines**, define test **approaches**
3. **Assess risks** and plan their **mitigation**
4. Define **Definition of Done and Definition of Ready** ☺
5. Set **workflow and artifacts**
6. Review and finalize strategy
7. Define metrics and KPI's

INITIATION

✔ **Contact** people (customer, PM, PO, Delivery Manager, Account Manager, QA Manager on customer side) and find out customers priorities (time-money-quality)

✔ Read **"Vision"** document or contact PM and analyze what kind of product will be developed, which quality parameters are important for it, what potential risks exist

✔ **Research** industry best practices

QUESTIONS FOR STAKEHOLDERS?

Which questions will you address to stakeholders?
- What is a project goal?
- Could you define what is the most important: quality, vs budget, vs timeline?
- What kind of product will be developed?
- Who are the end users of the product?
- Deadlines?
- Are other vendors participating? Geographical disposition of the teams?

QUESTIONS FOR STAKEHOLDERS?

Which questions will you address to stakeholders?
- What is the frequency of releases?
- Are requirements final? Who is responsible for requirements?
- Do you have a formal process to manage change requests?
- Who is responsible for environments management?
- Do you have defined success criteria?
- What is the escalation path?
- Tools?

ANALYSIS OF FUNCTIONAL SCOPE

Define test **coverage** and test **approaches** for:

End-to-
end
flows

Functional
areas
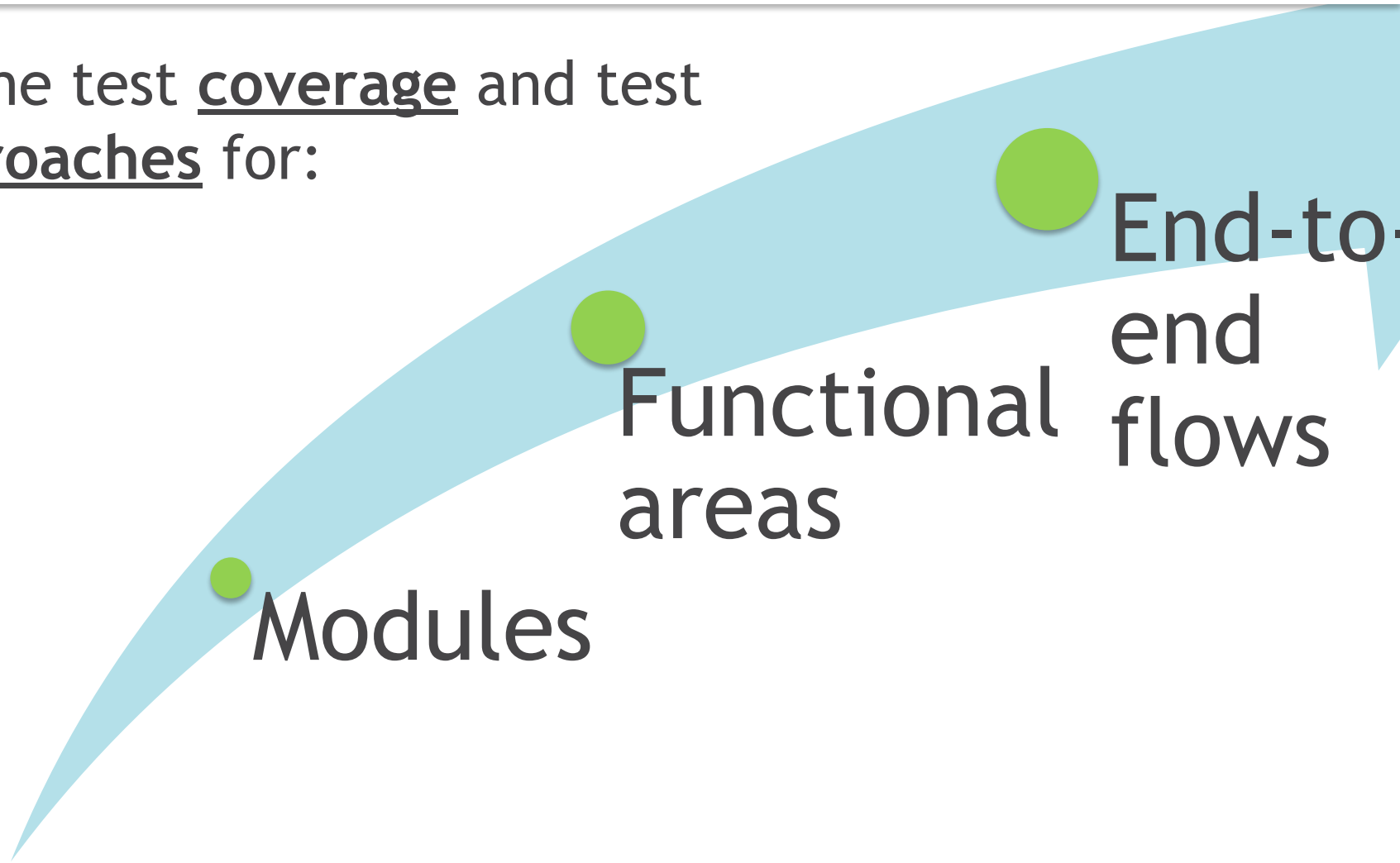
Modules

SCOPE

1. Identify application **modules**, their **interfaces**, **requirements** for them; define **modules test coverage** and **specific test types/approaches** for them
2. Split the **application into functional areas**, define **test coverage** with a **glance to modules coverage**
3. Define **business scenarios** and **full end-to-end flows** according to domain standards (full integration testing)

ANALYSIS OF NON-FUNCTIONAL SCOPE

When thinking about **test coverage** consider the following:

- Different configurations
- Localization
- Mobile impact
- Basic security vulnerabilities
- Performance (latencies, response time)
- Accessibility
- Other non-functional requirements

AUTOMATED TESTING

Consider **Automated testing:**

✔ Define what can be included in the automation test suite (**AT scope**)

✔ **Calculate ROI** in Automated Testing

✔ Define Automated Testing **tools**

- Be accurate with AT!
- Which advantages and disadvantages you see in AT?

AUTOMATED TESTING PROS

1. Customers like AT ☺

2. It is easy to sell (almost each Request For Proposal includes AT)

3. AT is a trend

AUTOMATED TESTING PROS

4. CI is impossible without AT (all advanced projects use CI) ☺

5. Speed up release

6. Human factor minimization

7. Save money?

AUTOMATED TESTING CONS

1. Expensive

2. Few bugs found

3. Much time for maintenance

AUTOMATED TESTING CONS

4. Trained specialists are needed

5. Instability of automated tests

6. Much time for run

AUTOMATED TESTING: WHY?

Before making decision about AT on your project, **ask customer why he wants automation**, the **goal** of it and how we will **understand that the goal is completed**.

Money
(economy on manual testing)

Time
(speed up release)

improve **Quality**

AUTOMATED TESTING: SAVE MONEY

Money
(economy on manual testing)

1. Calculate ROI (the more frequent releases the more effective AT)
2. Think over other ways of saving money on testing (e.g. stop testing not-risky areas)
3. Discuss the figures with customer before starting AT

AUTOMATED TESTING: WASTE MONEY

Money
(economy
on manual
testing)

ROI: AT investment may not be occupied!

Example: Supply-On project.

AUTOMATED TESTING: SHORTEN TIME

Time
(speed up release)

1. Learn the target frequency of releases

2. Implement **CI**

3. Increase the scope for API and Unit Tests instead of UI tests

AUTOMATED TESTING: SHORTEN TIME

Time
(speed up release)

4. Speed up run time (run different UI tests in parallel)

5. Create more stable tests (use effective tool, framework; collaborate with developers to make UI easy to automate)

AUTOMATED TESTING: TIPS & TRICKS

Time
(speed up release)

1. If releases (Potentially Shippable Increments) are not frequent, it does not make sense to implement AT
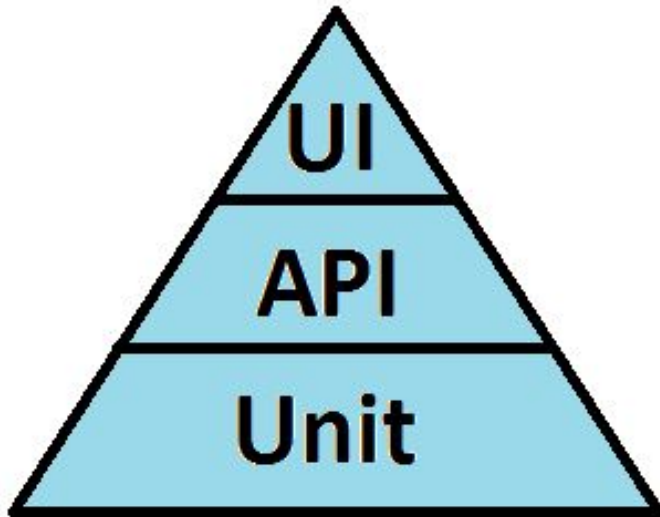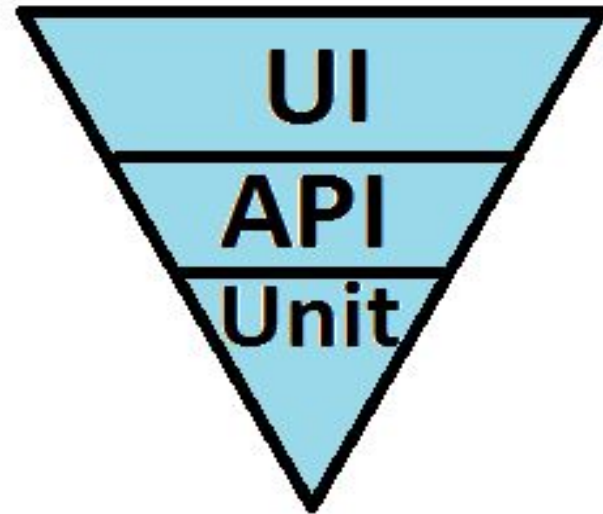
Time
(speed up release)

1. Effective tool- **make comparative table of pros and cons of each tool**, verify AT scrips of existing tool, check if existing scripts find bugs. General recommendations: **Selenium** WebDriver for **Web**, MS VS **CodedUI** for **Desktop**, **Appium** for **Mobile** and **Desktop**.

AUTOMATED TESTING: UNIT, API, UI

Ideal

Customer wants

UI automation works with browser, not with code. Unit tests are more effective: easier to create, faster to run, more stable than UI tests.

AUTOMATED TESTING: IMPROVE QUALITY

improve
Quality

1. Think carefully if AT is the right decision to improve quality

2. **Calculate metrics, perform project assessment**

3. Elaborate improvements

4. **Communicate improvements to customer**

RISKS DEFINITION

✔ Define testing **risks** and **mitigation plan** (**who, when, what should do with them**);

✔ Set the **priorities** for the functional areas from end-user point of view;

RISKS DEFINITION

✔ Organize test cases according to the functional areas and priorities (the highest risk should be tested firstly)

✔ Define **regression strategy** based on risk analysis;

✔ Make rough **estimation** for each testing type

RISKS EXAMPLES

RISKS EXAMPLES

✔ Acceptance Criteria for User Stories are not complete

✔ Test Data not provided in time

✔ Test environment is not available for testing in time

✔ Impossible to perform integration testing because integration with third party systems is not tuned

✔ Scope of changes is huge

EXAMPLE OF ROUGH ESTIMATIONS

Test Types	Area	Risks (%)	Balance (PD)
new feature	area1	23	280
new feature	area2	37	440
regression		20	240
localization		9	108
compatibility		3	36
performance		5	60
security		2	24
installation		0	
mobile		1	12
Total		100	1200

EXAMPLE OF ROUGH ESTIMATIONS

- You have a deadline, you analyzed the scope-> you need to provide solution, how to test with maximum value and how many resources (testers) are needed.

- On the example: to perform selected testing types within 1 year, you need 1200 person-days (5 testers).

- Or you may decrease the quantity of testers and reduce either types of testing or time for new feature testing.

EXAMPLE

DEFINITION OF DONE

✔ Define testing **entrance criteria** inline with release milestones (code freeze, feature freeze and others)

✔ Discuss and agree on **Definition of done (DoD)** with stakeholders

EXAMPLE OF DOD

Example of DoD:

✔ The sprint is closed after new feature and regression testing, it is completed when there are no open Blocker/Critical bugs.

✔ User stories should be accepted only after dev + test + fix + retest phases

✔ All found reported bugs are linked to related user story

Example of entrance criteria:

1. Each user story has acceptance criteria, approved by PM/customer before sprint start.

2. Code freeze is done <N> days before the end of sprint.

3. Additional code check-ins are allowed only for fixes of critical/major bugs.

WORKFLOW AND ARTIFACTS

✔ Define testing **tools**;

✔ Define testing **environment**;

✔ Describe actions testers should perform during agile **process** in terms of one sprint;

✔ Describe **bug workflow**;

WORKFLOW AND ARTIFACTS

✔ Define template for **reporting** and its frequency;

✔ Create **communication plan**;

✔ Define and describe **review** process for test artifacts (test cases, defects);

✔ Define **quality metrics** and the way to track them.

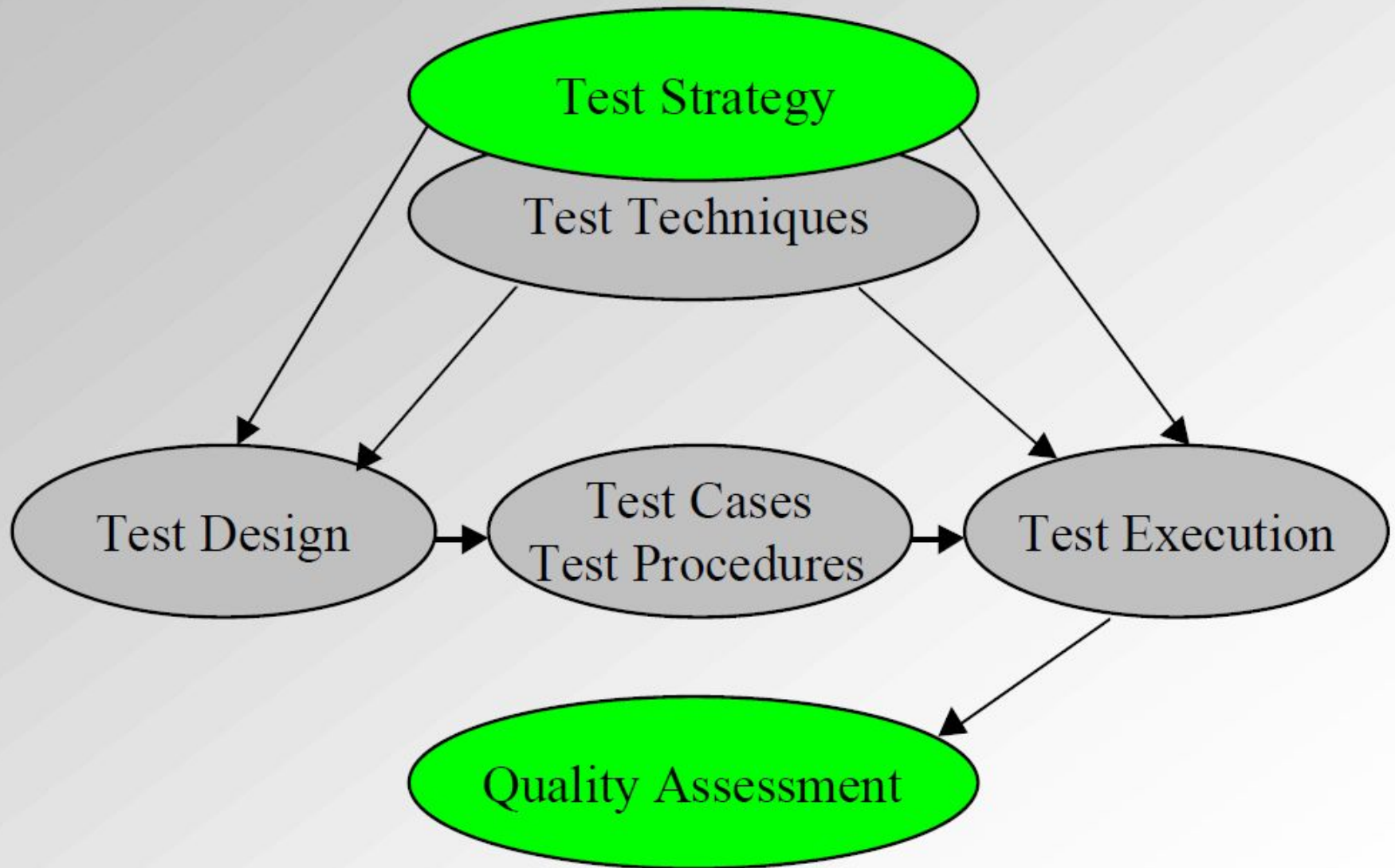# REVIEW AND FINALIZE

✔ Make **peer review** of test conditions – discuss test coverage with developers

✔ Discuss Test Strategy with customer and other stakeholders

✔ **Finalize** Test Strategy

TEST STRATEGY EXECUTION

After execution of Test Strategy we should constantly **optimize** it and **make improvements.**

Let's talk about it the next time ☺️

WHAT WE LEARNED

- Our goal is to provide **testing** as a **value-added Service**

- To **bring value** to customer we should **identify** his **goals**, **priorities**

- Customer cares about **time, money, quality, visibility**

- To meet customer goals we should **elaborate effective test strategy**

WHAT WE LEARNED

- Effective Test Strategy is first of all a **thought process, not a document**
- Effective Test Strategy defines:
  - Approaches
  - Risks
  - Processes
  - Definition of Done and Definition of Ready
  - Tools
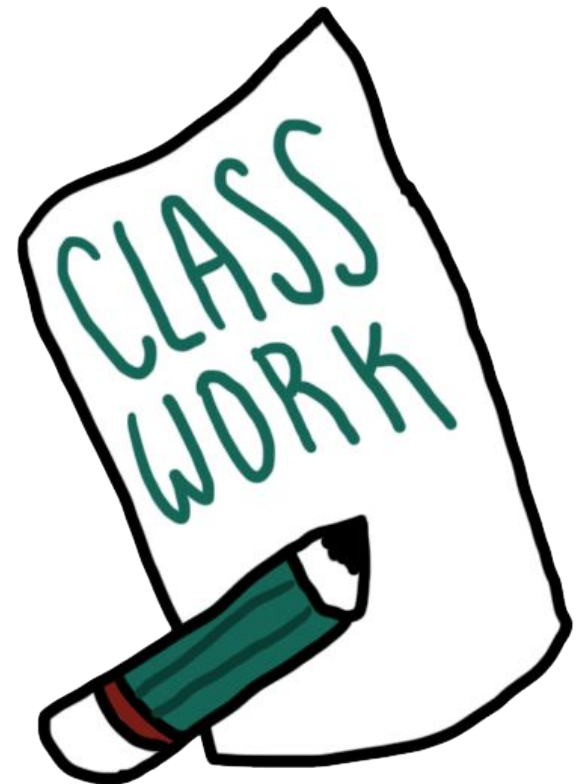 - KPIs: metrics, their target value, frequency of calculation

CLASSWORK ☺

CLASSWORK ☺️

1. You'll be provided with description of projects.
2. You need to elaborate Test strategy on these projects
   - Risks
   - Approaches
   - Specificity

SITUATION #1

- Web-Portal for selling goods.
- Targeted consumers are in US, Canada, UK, Russia, EU countries, China, Japan, Arab Emirates.
- Ability to search goods, check availability in warehouses, booking, delivery and integration with range of payment systems.
- Mobile version with all devices support should be available for users.

SITUATION #1

- A team of QA engineers on customer side from Tokyo and New York offices will be involved into testing process together with our testing team.
- Our Test TeamLead should plan and coordinate all testing activities on the project except UAT stage.
- Project duration is 1 year. After product rollout, support phase with CRs development and bug fixing is planned.
- Customer proposes to use IBM Jazz tool for bug-tracking and test management purpose as he got good feedbacks about IBM tools.

SITUATION #2

- Movement of existing corporate Product Data Collection (PDC) service to a new platform
- When a new product appears in corporate product catalog, PDC service starts collecting data related to product (description, ratings, reviews etc.) from 25 <u>free</u> internal services and 16 external <u>paid</u> services on weekly basis, then analyzes collected information about product and updates data in DB correspondently.

SITUATION #2

- Corporate Report Portal and Business Development tool use this product data for their needs.
- Customer has bad experience using Agile methodology.
- Corporate Report Portal is under development by other IT vendors.
- Business Development tool is by now migrated to new platform.
- PDC and report Portal have the same roll-out deadline.

SITUATION #3

- Development of a new back-end system
- System should support handling of huge range of financial transactions in the 24\7 mode
- Fixed product rollout deadline.
- Non-functional requirements and acceptance criteria for them will be specified at the beginning of the project

YOUR SITUATIONS

Recommended videos EN

- Rikard Edgren "An Introduction to Test Strategy"
https://www.youtube.com/watch?v=OZiE9eApOXY

- Testing Strategies for Continuous Delivery
https://www.youtube.com/watch?v=DgQWSaCQ82U

- Huib Schoots "Practical Test Strategy Using Heuristics"
https://www.youtube.com/watch?v=_TE9RFzNs_M

- Paul Gerrard "Agile Test Strategy"
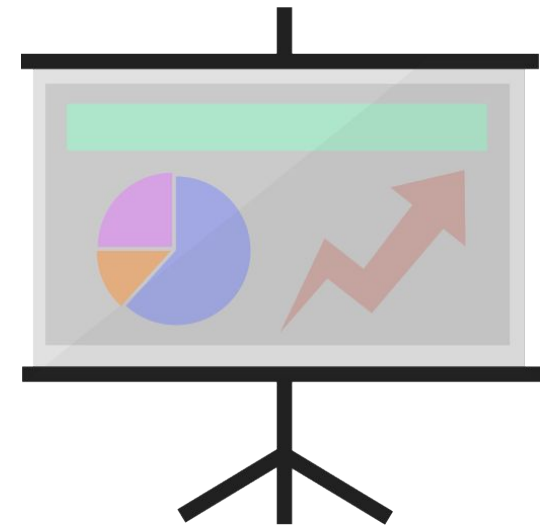https://www.youtube.com/watch?v=Ed6YkYEkCRM

Recommended videos RU

- Сергей Мартыненко "Подготовка стратегии тестирования под высокорискованный, высокодоходный проект"
  https://www.youtube.com/watch?v=z3NuJb0_MzI
  Note: IMHO too complicated \ too theoretical for most of us

- Radio QA "Выпуск 10: Стратегия тестирования»
  http://radio- qa.com/vypusk-10-strategiya-testirovaniya/
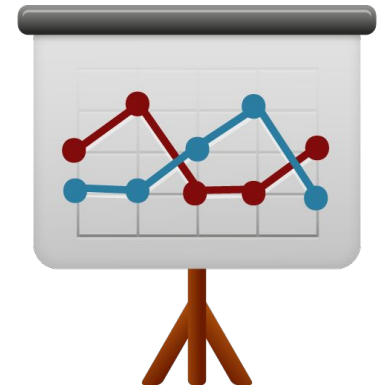 *Note: IMHO too theoretical for most of us*

Co-presentations

- Testing Metrics fundamentals.pdf
- Testing Metrics.pdf

- QA Automation ROI - general information.pdf
- QA Automation ROI Calculator (Ru).pdf
- Startup test automation on the project (En).pdf

- Test Pyramid vs ROI SQA Days.pdf

Trainings

- http://dpi.solutions/education?name=**testing-strategy**

- http://dpi.solutions/education?name=**roi-for-automation-testing**

- http://dpi.solutions/education?name=**metrics-in-testing**

CONTACT ME

✉ **Semenchenko@dpi.solutions**

Ⓢ **DPI.Semenchenko**

in **https://www.linkedin.com/in/anton-semenchenko-612a926b**

f **https://www.facebook.com/semenchenko.anton.v**

🐦 **https://twitter.com/comaqa**