Amplifying Agile Quality Practices

SELENA DELESIE

@SelenaDelesie

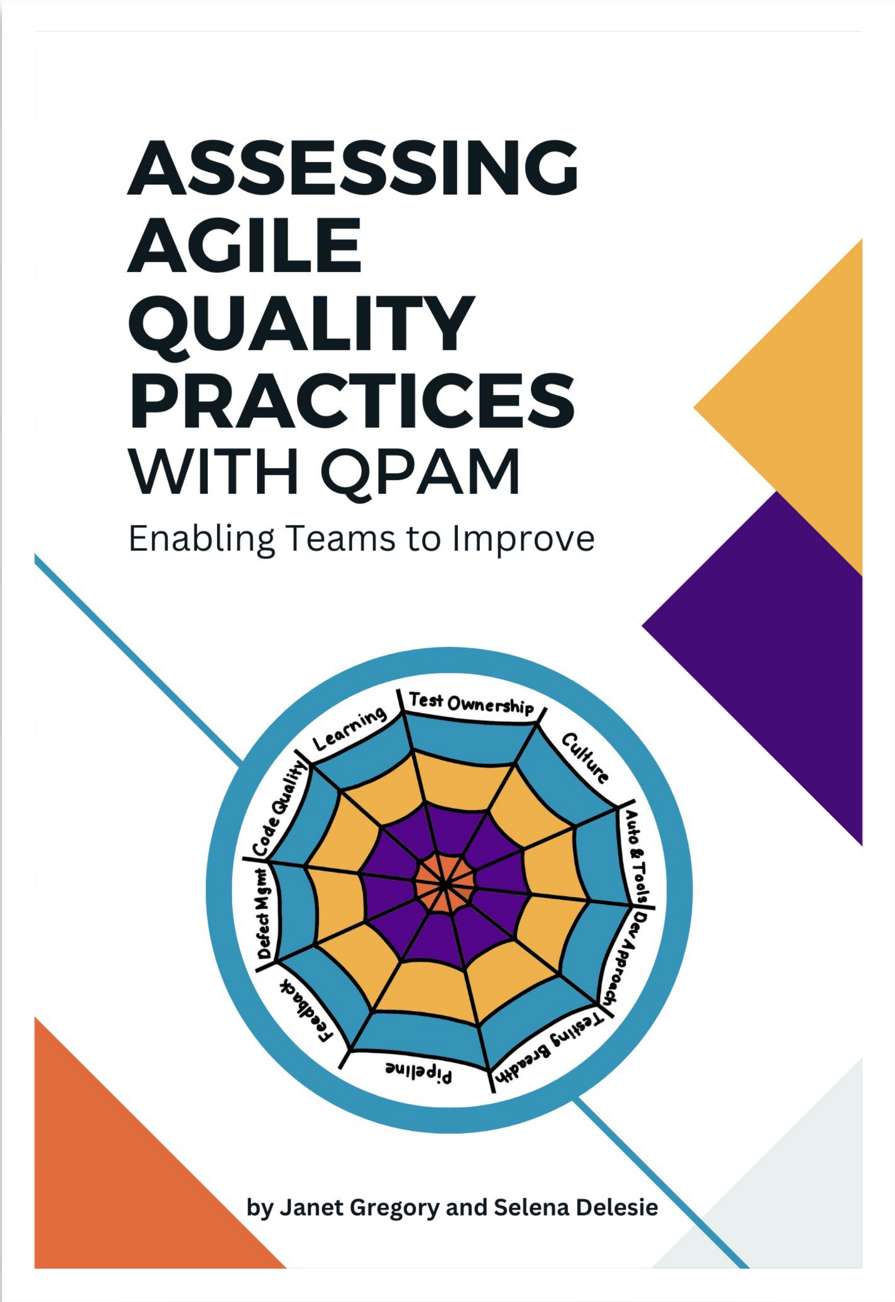
Selena@AtmaGroup.ca

Selena Delesie

Atma Group Inc.

Coach | Speaker | Author | Teacher

@SelenaDelesie

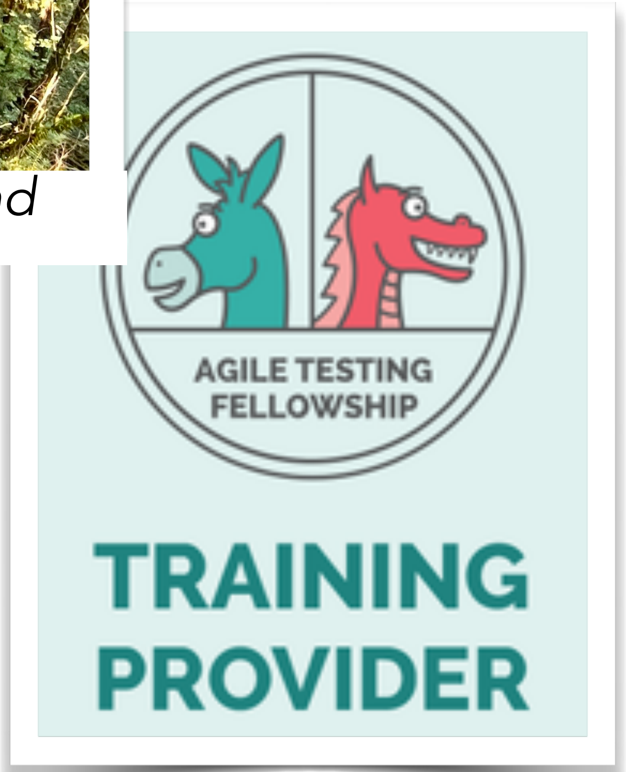


ASSESSING AGILE QUALITY PRACTICES WITH QPAM

Enabling Teams to Improve

by Janet Gregory and Selena Delesie

Hiking Marquam Nature Park, Portland

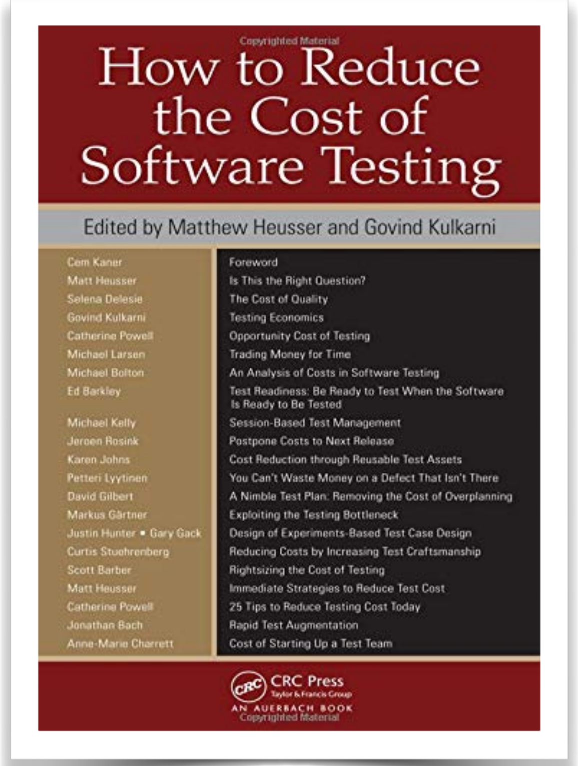


AGILE TESTING FELLOWSHIP

TRAINING PROVIDER

Tips From The agile Trenches

Collected by @YvesHanoulle



How to Reduce the Cost of Software Testing

Edited by Matthew Heusser and Govind Kulkarni

Cem Kaner	Foreword
Matt Heusser	Is This the Right Question?
Selena Delesie	The Cost of Quality
Govind Kulkarni	Testing Economics
Catherine Powell	Opportunity Cost of Testing
Michael Larsen	Trading Money for Time
Michael Bolton	An Analysis of Costs in Software Testing
Ed Barkley	Test Readiness: Be Ready to Test When the Software Is Ready to Be Tested
Michael Kelly	Session-Based Test Management
Jeroen Rosink	Postpone Costs to Next Release
Karen Johns	Cost Reduction through Reusable Test Assets
Petteri Lyytinen	You Can't Waste Money on a Defect That Isn't There
David Gilbert	A Nimble Test Plan: Removing the Cost of Overplanning
Markus Gärtner	Exploiting the Testing Bottleneck
Justin Hunter • Gary Gack	Design of Experiments-Based Test Case Design
Curtis Stuehrenberg	Reducing Costs by Increasing Test Craftsmanship
Scott Barber	Rightsizing the Cost of Testing
Matt Heusser	Immediate Strategies to Reduce Test Cost
Catherine Powell	25 Tips to Reduce Testing Cost Today
Jonathan Bach	Rapid Test Augmentation
Anne-Marie Charrett	Cost of Starting Up a Test Team

CRC Press
Taylor & Francis Group
AN AUERBACH BOOK
Copyrighted Material

41ST ANNUAL
PACIFIC NW SOFTWARE QUALITY CONFERENCE
OCT. 9-11,2023

Agile Values… Intended

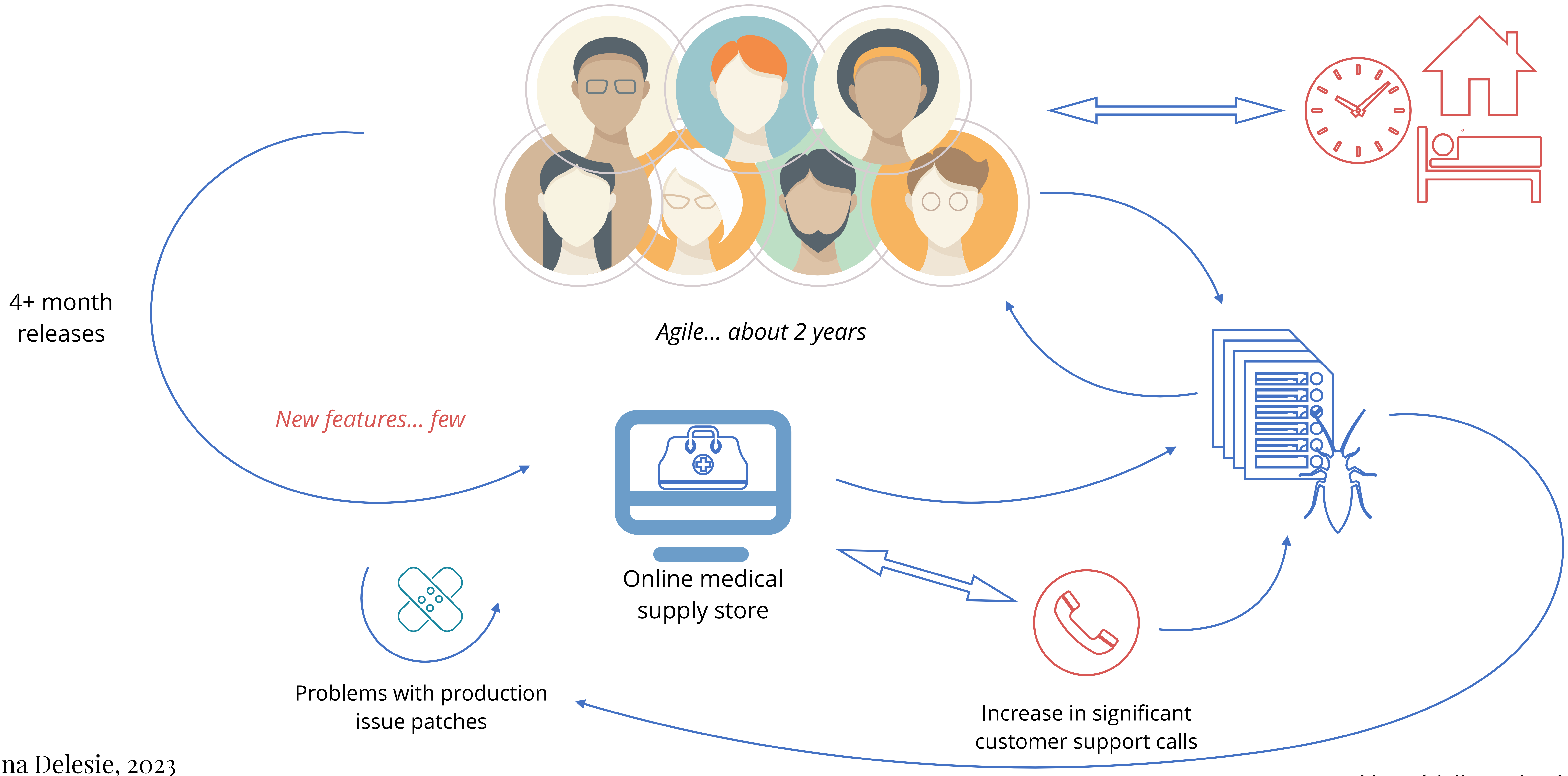
Individuals and Interactions

Customer Collaboration

Working Software

Responding to Change

Meet Team Venus



Agile… about 2 years

4+ month releases

New features… few

Online medical supply store

Problems with production issue patches

Increase in significant customer support calls

41ST ANNUAL PACIFIC NW SOFTWARE QUALITY CONFERENCE OCT. 9-11,2023

Amplifying Agile Quality Practices

Image by Arek Socha from Pixabay

“Without quality, there is no value”

~ Selena Delesie

Image by Gerd Altmann from Pixabay

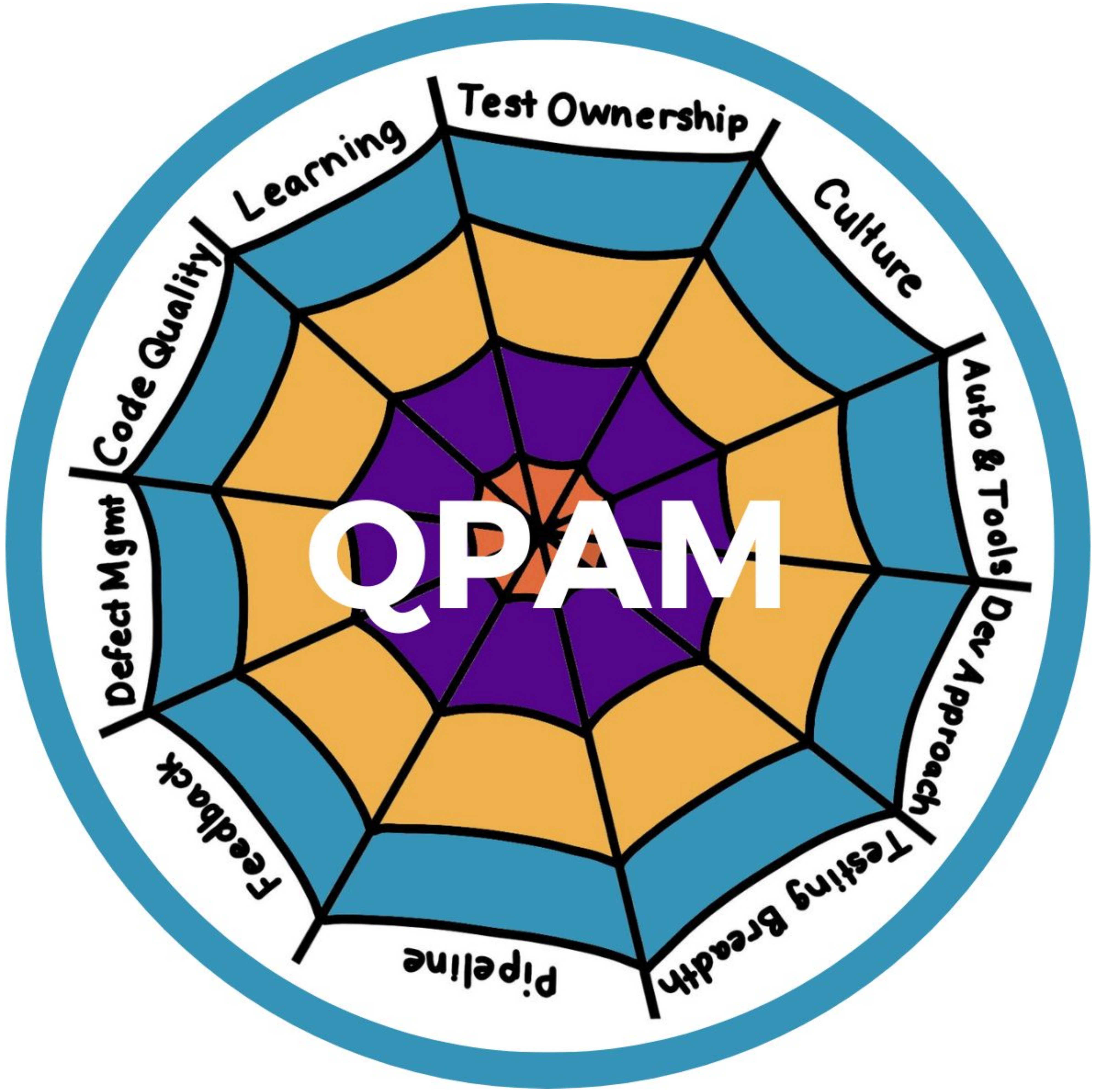
A tool to discover what is going on

QPAM
Quality Practices Assessment Model

A tool for teams to reflect on or assess the quality of their practices and processes.

Co-created with Janet Gregory

Inspired by Alan Page's Quality Culture Transition Guide



The Ten Quality Aspects

1. Feedback loops
2. Culture
3. Learning and improvement
4. Development approach
5. Quality and test ownership
6. Testing breadth
7. Code quality & technical debt
8. Test automation and tools
9. Deployment pipeline
10. Defect management

The Social Quality Aspects

Feedback loops: how the team communicates (within the team, with customers, with leadership)

Culture: psychological safety, quality culture, working hours

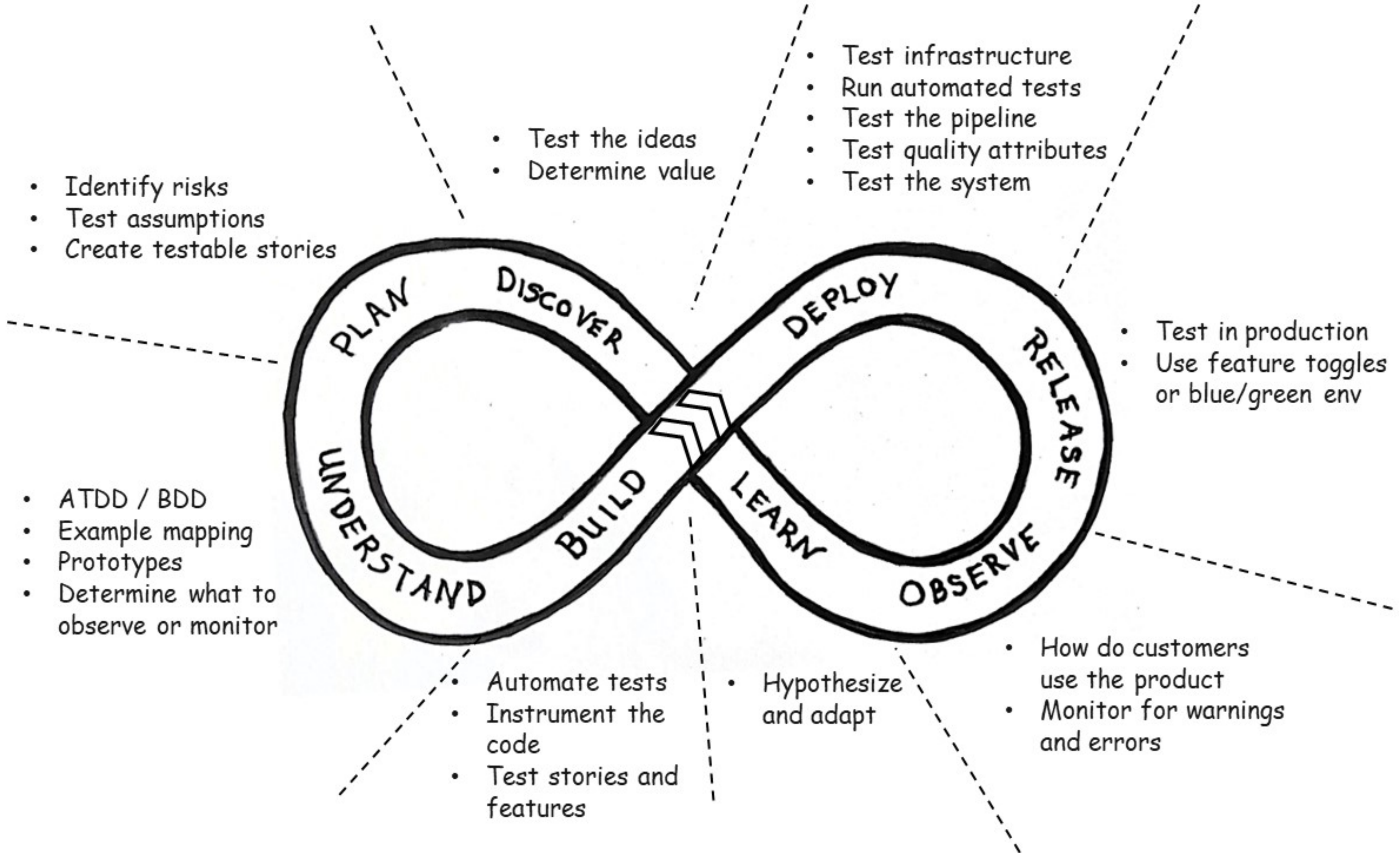
Learning and improvement: self-directed learning, conferences, sharing, retrospectives

Revisit Team Venus

Feedback Loops	• Working with some structure, using scrum • Sometimes discuss user stories with the customer and management • Have monthly demos with the customer • Team members usually work independently and report their status at the daily standup
Culture	• Management sets quality directives • All team members regularly work overtime • People do not speak to management about problems
Learning + Improvement	• Team members usually able to attend a training or conference yearly • Do retrospectives monthly at best, and make some improvements • Some significant defects have root cause analysis done to understand the problem

Social and Technical Combined Quality Aspects

- **Development approach:** team makeup, quality mindset, agile framework practices, feature and story prioritization, story understanding, Definition of Done, release to production.
- **Quality and test ownership:** quality understanding, creating and executing tests, following up on test failures as well as data ownership.



- Identify risks
- Test assumptions
- Create testable stories

- Test the ideas
- Determine value

- Test infrastructure
- Run automated tests
- Test the pipeline
- Test quality attributes
- Test the system

- Test in production
- Use feature toggles or blue/green env

- How do customers use the product
- Monitor for warnings and errors

- Hypothesize and adapt

- Automate tests
- Instrument the code
- Test stories and features

- ATDD / BDD
- Example mapping
- Prototypes
- Determine what to observe or monitor

Revisit Team Venus

Development Approach	• Lack of good communication or early collaboration about user stories • Programmers often decide what a story means and code accordingly • Team members tend to work in silos and independently • Testing often happens after coding is completed for each story • Production releases happen every 4-6 months, with many issues
Quality & Test Ownership	• Team discusses how to achieve quality directives together • Programmers sometimes create unit tests, and integration tests • All team members discuss functional tests and how best to create them • Programmers help automate some functional tests, when there is time

The Technical Aspects

Testing breadth: test strategy, test techniques, testing quality attributes, UAT, test reporting and coverage

Code quality and technical debt: unit testing, code maintenance, standards

Test automation and tools: automating functional tests, quality attribute tests, and unit tests; tools

Deployment pipeline: automation in testing and infrastructure; team engagement in the process, release strategy, metrics, observability

Defect management: reporting, fixing, metrics and defect triages

Revisit Team Venus

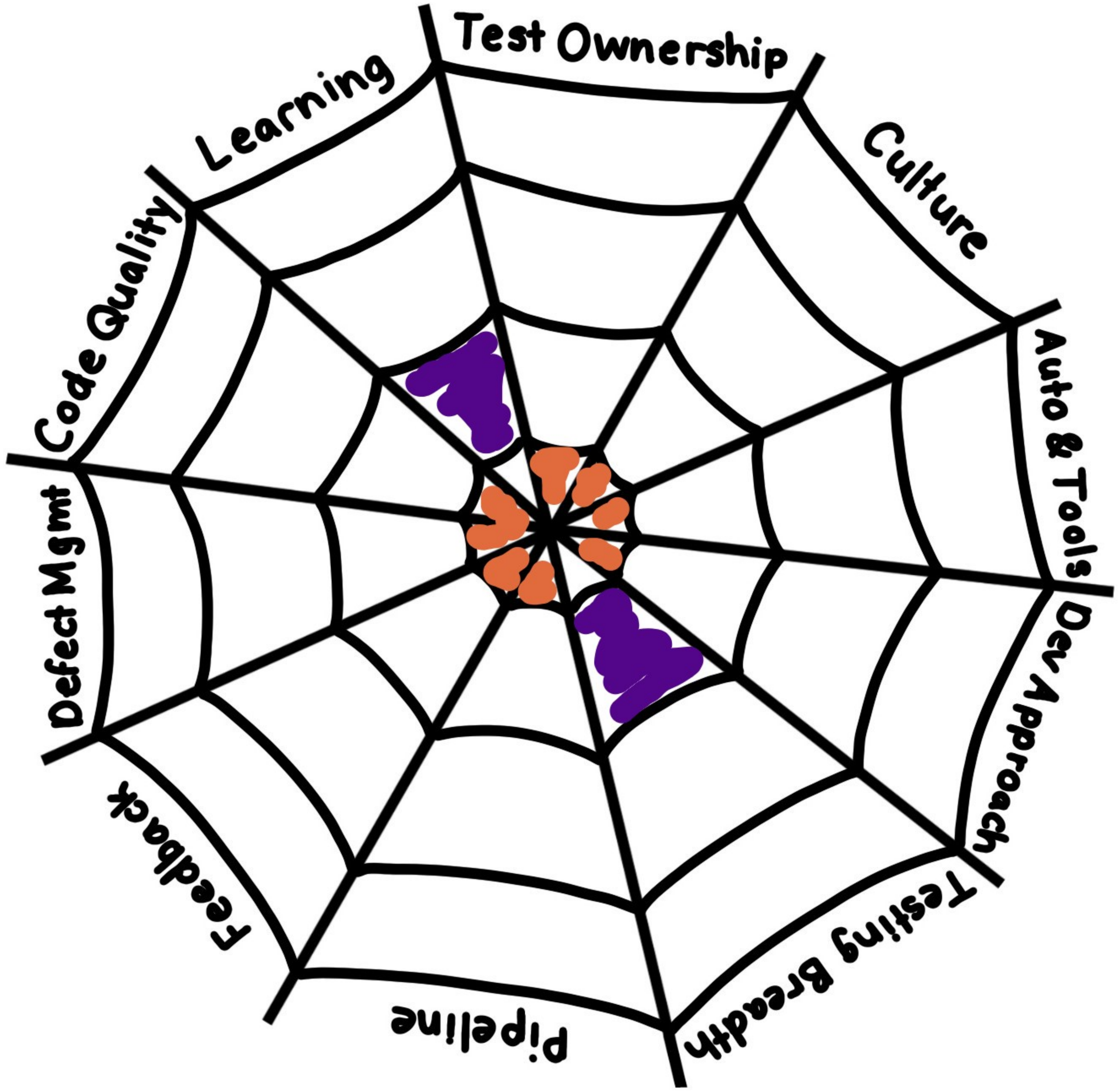
Testing Breadth	• whole team discusses test strategy upfront, testers usually create it • team employs different test techniques and consider quality attributes • team plans using testing quadrants and test automation pyramid
Code Quality & Technical Debt	• have coding standards • using code quality and analysis tools to identify gaps • team has a backlog of technical debt items to address
Test Automation & Tools	• team automates new feature functional tests • small set of unit tests that are slowly being added to for new features, defect fixes, and patch releases
Deployment Pipeline	• starting towards continuous integration • often have problems with test environments • automated tests starting to use version control
Defect Management	• shifting to defect prevention • defect backlog is big, most are known for 3+ months • have regular defect triage meetings to identify which to fix for releases

The Four Dimensions

1. Beginning
2. Unifying
3. Practicing
4. Innovating

It is not a maturity model, but a starting point for where teams are in their journey and what the possibilities might be.

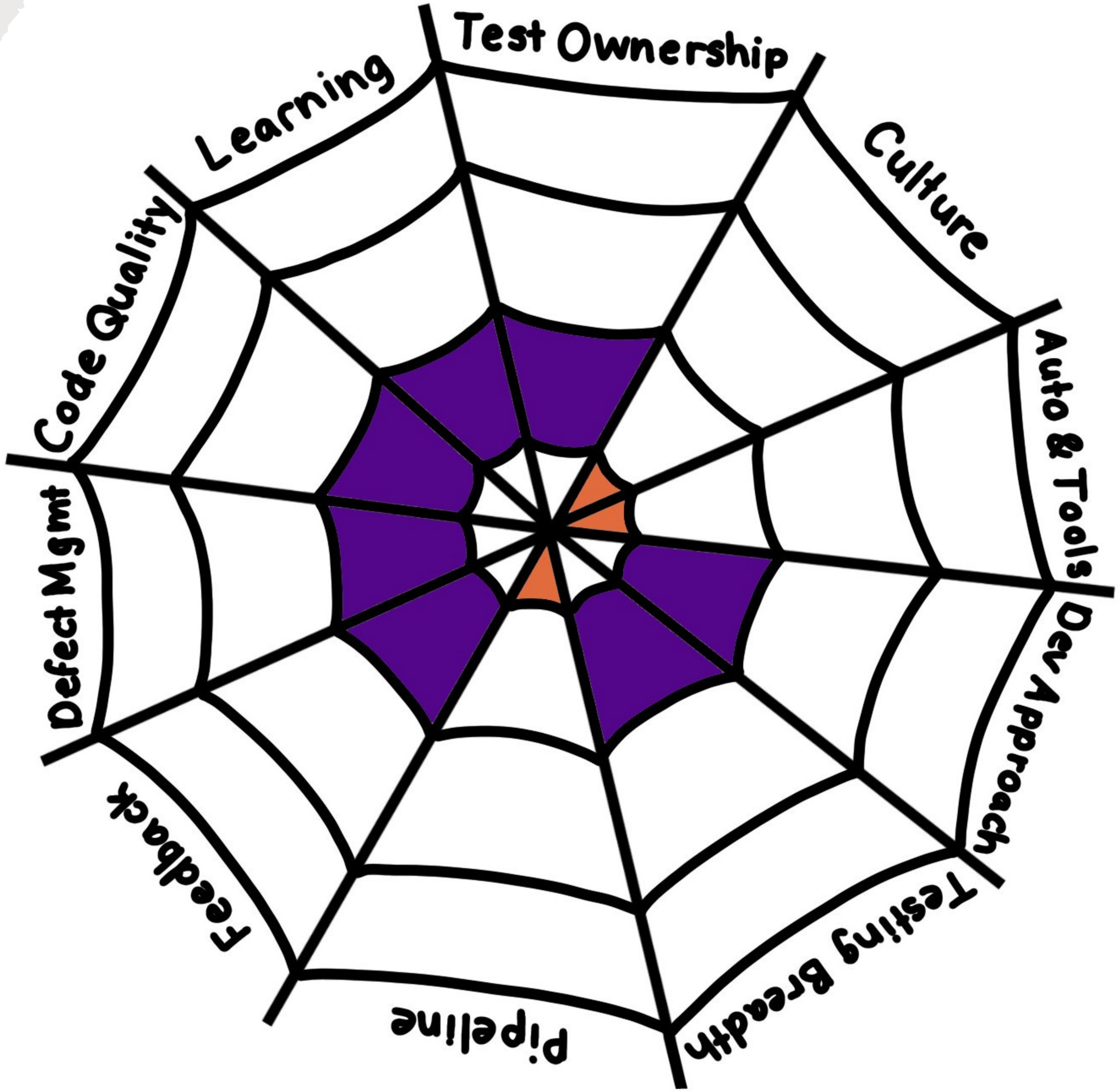
Beginning Dimension

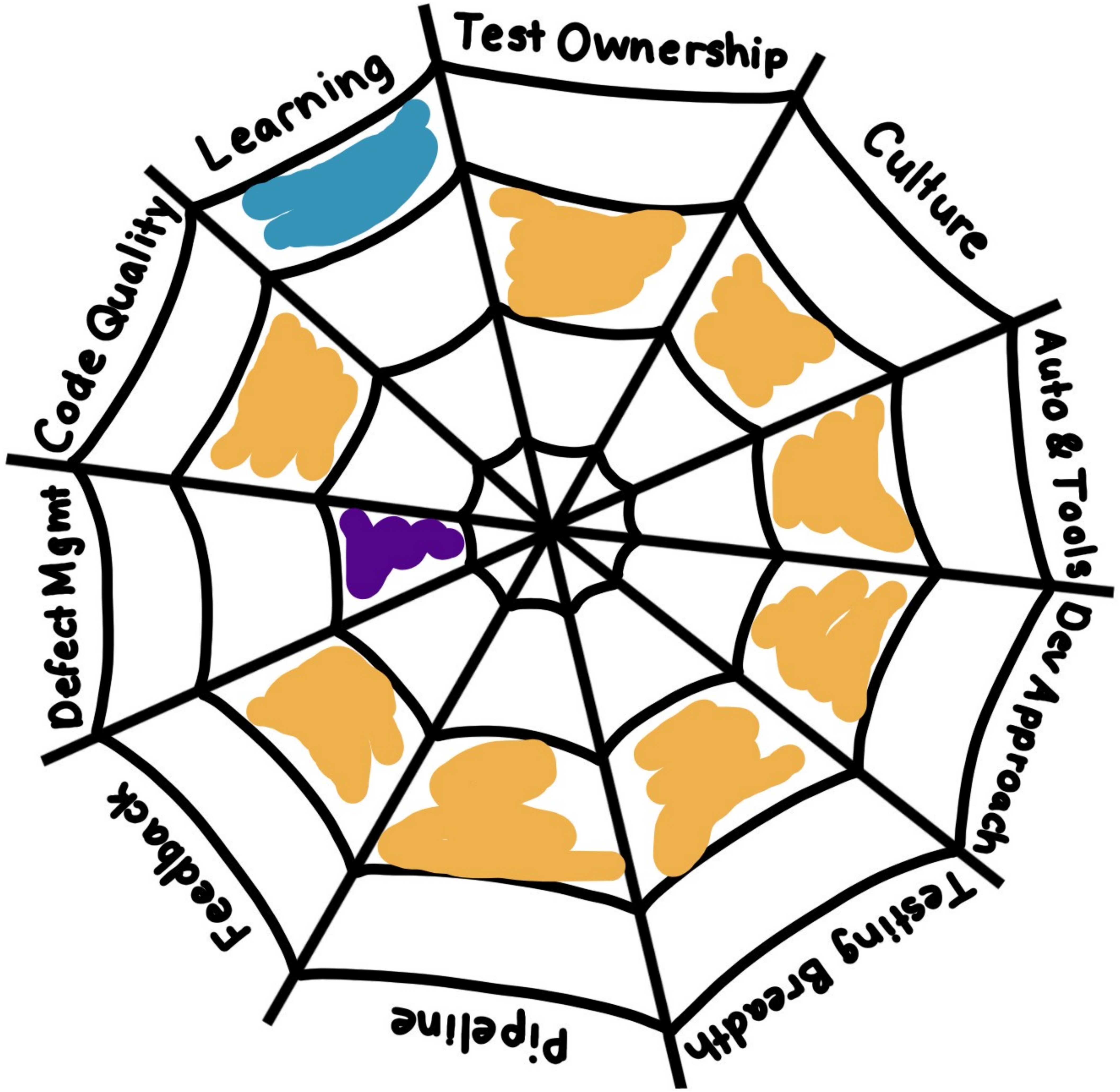


- Little structure, few processes
- Work in silos
- Low quality code
- Many defects
- Little to no automation
- May feel like chaos

Unifying Dimension

- Adoption of one or more agile methods (Scrum / XP / Kanban)
- Cross-functional teams
- Learning to work is smaller chunks
- Starting to release more often
- Test automation is stressed



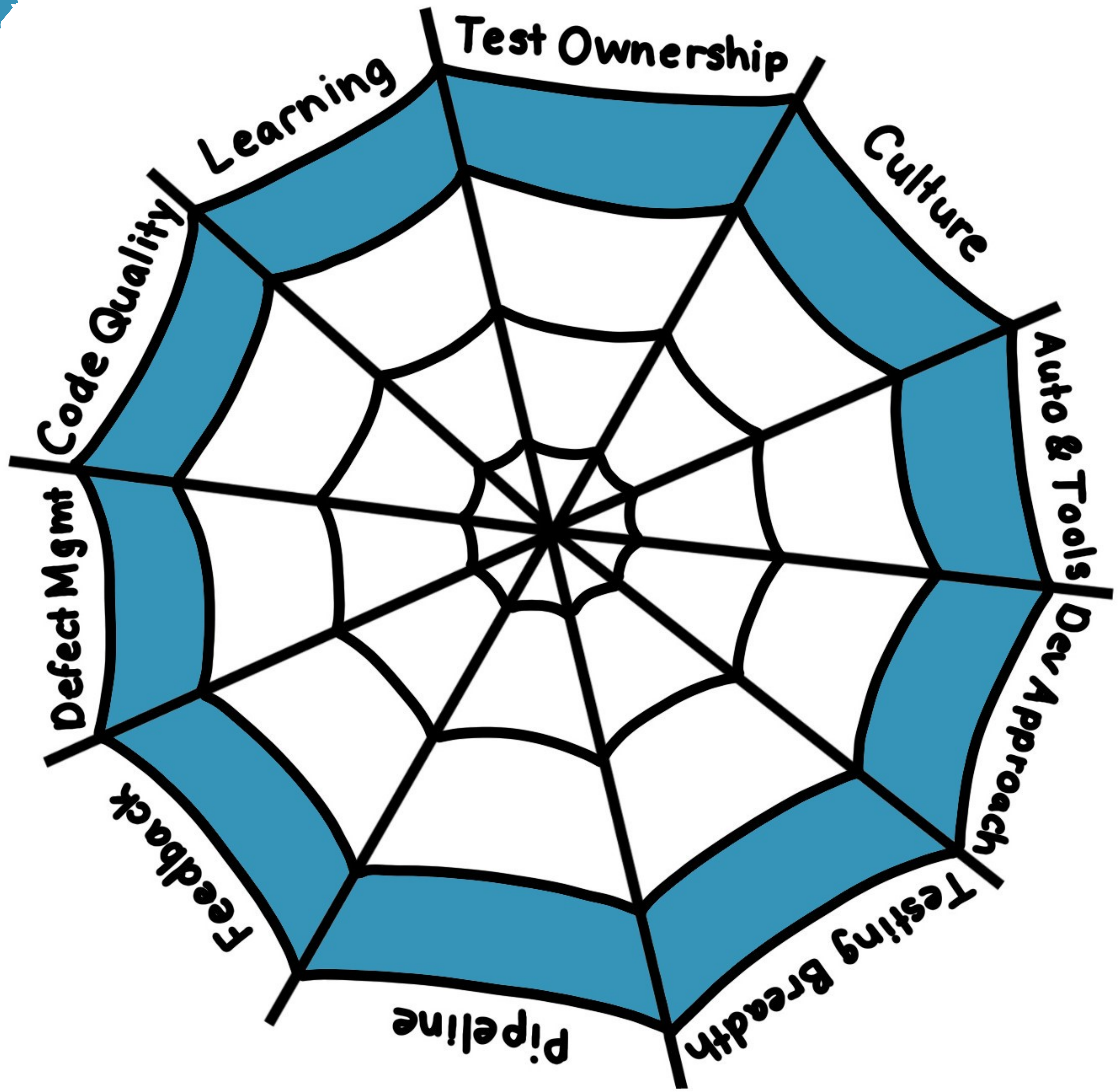


Practicing Dimension

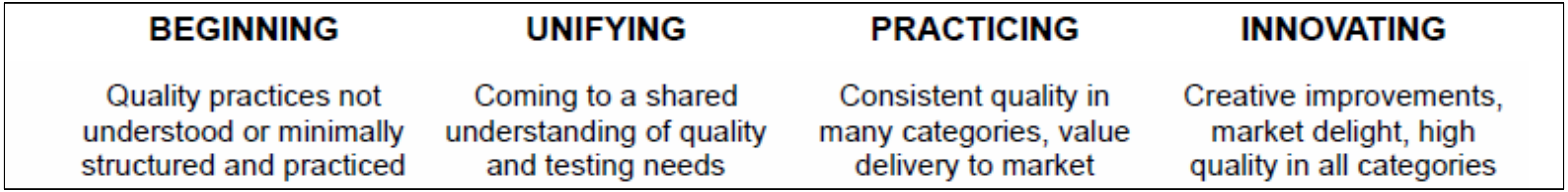
- Practices feel natural
- Consistently deliver value to their customers
- Fast effective feedback loops
- Processes and structure focus on quality and flow
- Think about testing holistically

Innovating Dimension

- High performing team
- Short cycle times, delivering with high quality
- Self-disciplined, striving to learn and improve
- Uses monitoring and observability to learn
- Quality is built-in from the start



The DIMENSIONS



BEGINNING	UNIFYING	PRACTICING	INNOVATING
Quality practices not understood or minimally structured and practiced	Coming to a shared understanding of quality and testing needs	Consistent quality in many categories, value delivery to market	Creative improvements, market delight, high quality in all categories

The QUALITY ASPECTS

Social aspects

1. Feedback loops
2. Culture
3. Learning & improvement

Social & Technical combined

4. Development approach
5. Quality and test ownership

Technical aspects

6. Testing breadth
7. Code quality & tech debt
8. Test automation and tools
9. Deployment pipeline
10. Defect management

Facilitating an Assessment

- Process retrospectives, interviews, observation, artifacts
- Prepare your questions
- Listen to understand

Safety for team members and trust in the facilitator are key for a successful assessment.

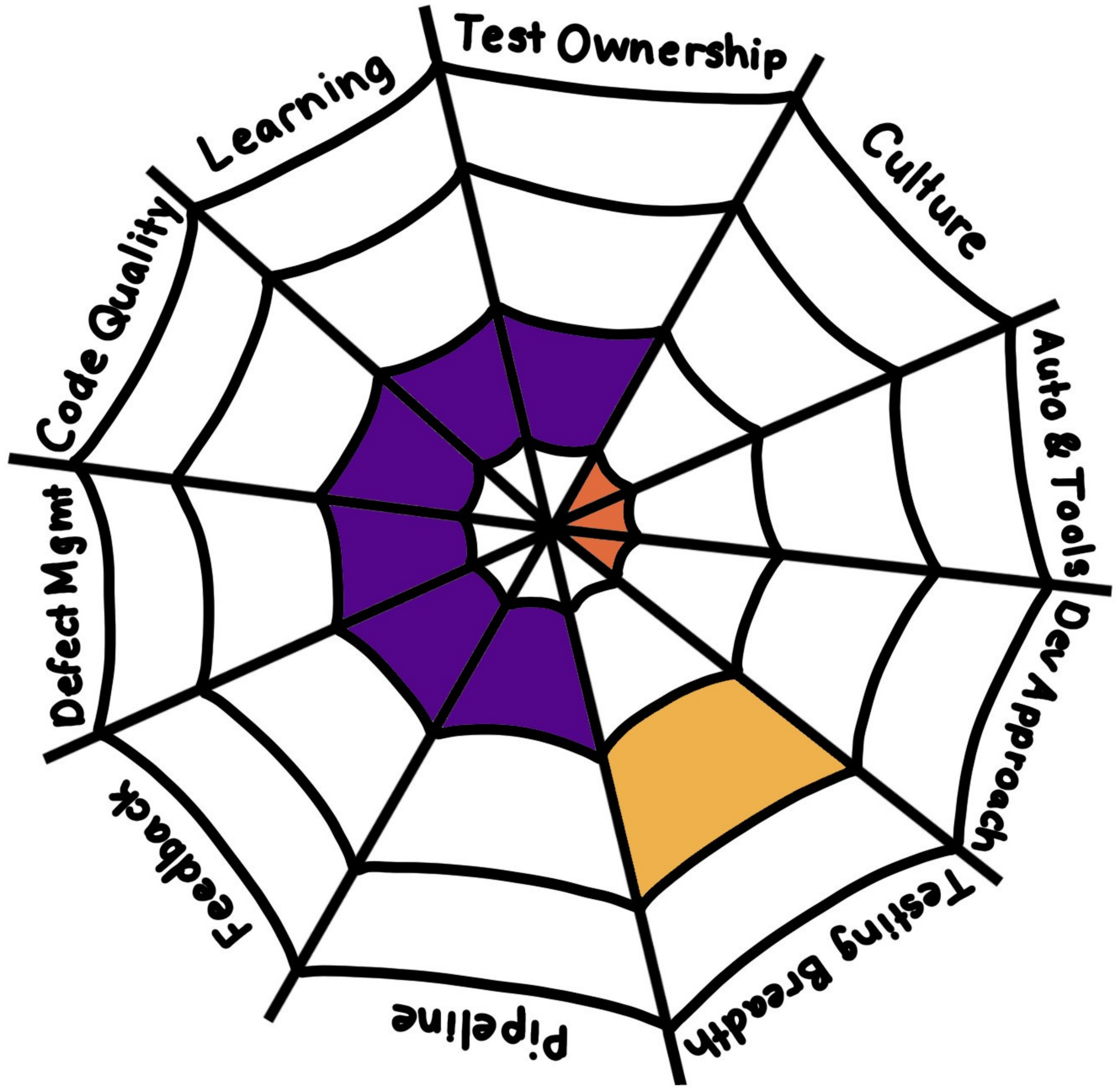
Examples of the Questions

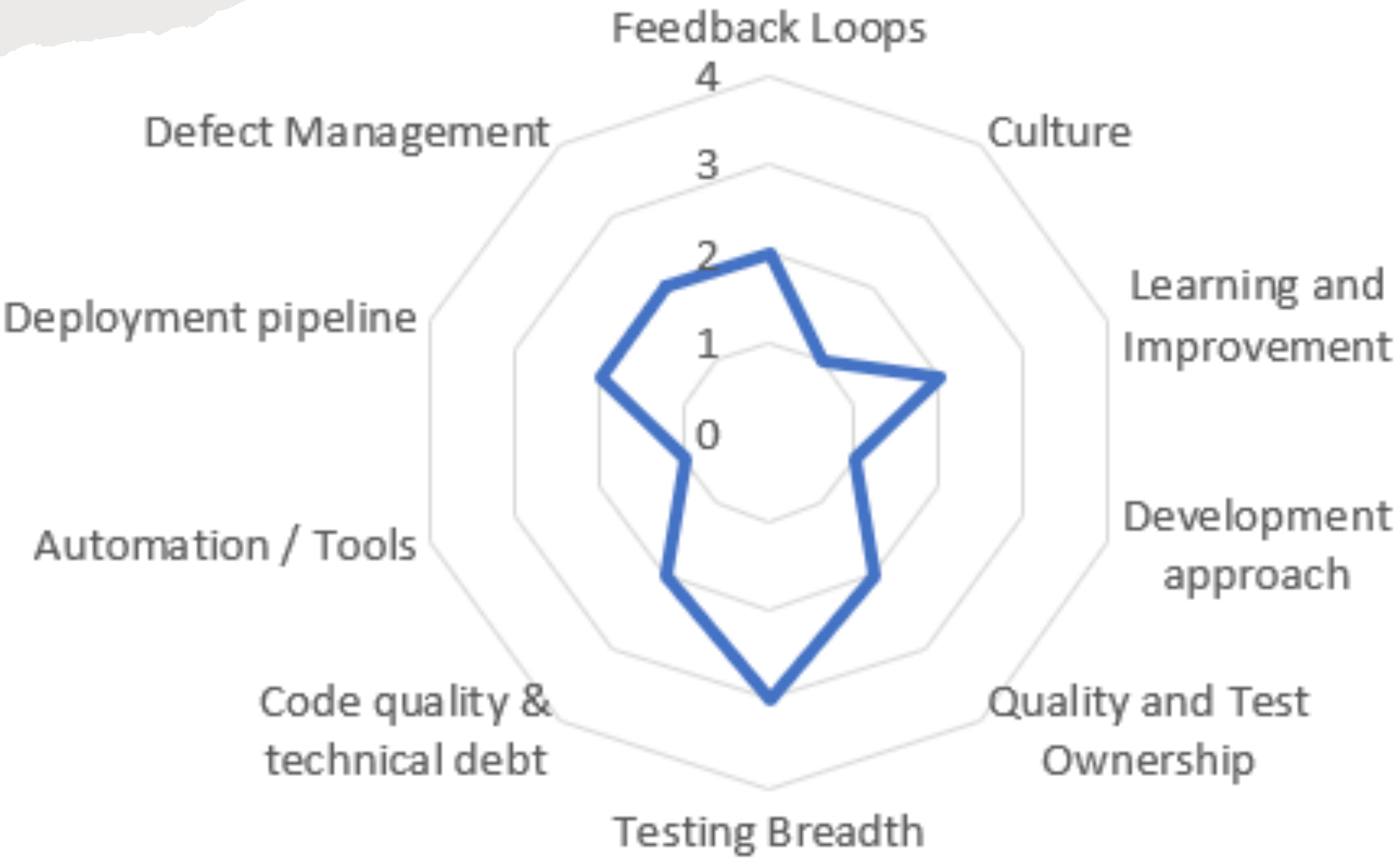
- **Feedback loops** - What mechanisms do team members use to communicate with each other?
- **Culture** - How are problems and challenges discussed and dealt with?
- Working hours - What is the office culture expectation and practice of working at a sustainable pace?
- **Learning and improvement** - How do team members share new practices, ideas, and skills?
- **Dev approach** - How do the team members work together?
- **Quality and test ownership** - How does the team determine what quality means for their product?
- **Testing breadth** - How do they determine what to test? How do they plan?
- **Code quality -** How does the team ensure coding standards are followed?
- **Test automation** - Does the team have an automation strategy?
- **Deployment pipeline** - What automated tests run in which environment?
- **Defect management** - What metrics are used for defect tracking? How are they used?

Putting it together

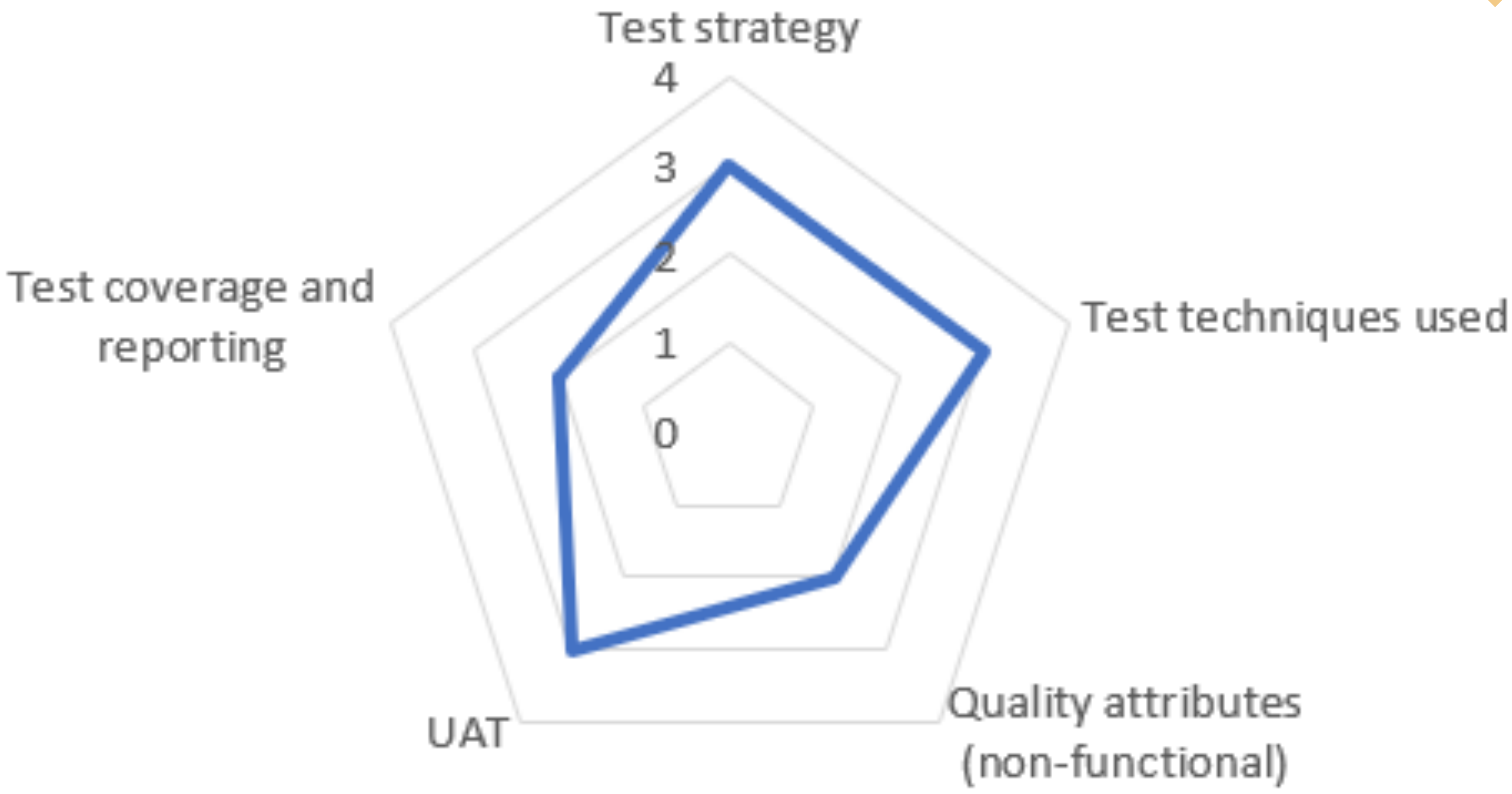
- Consolidate the information
- Set clear expectations
- Internal assessment: provide observations, teams decide what to improve
- External facilitator: provide observations, suggestions, recommendations
- Radar chart – one way to present the information

Team Venus
Example Radar Chart





Testing Breadth



Team Venus - Goals

- Make communication, collaboration, and a high-quality work ethic standard
- Reduce defects
- Improve time-to-market
- Higher-quality releases
- Have team members enjoy spending time on fun and value-adding activities
- Result in management delighted about reduced costs

Team Venus:
Possible Choices to Amplify Agile Quality Practices

- **Feedback Loops**:

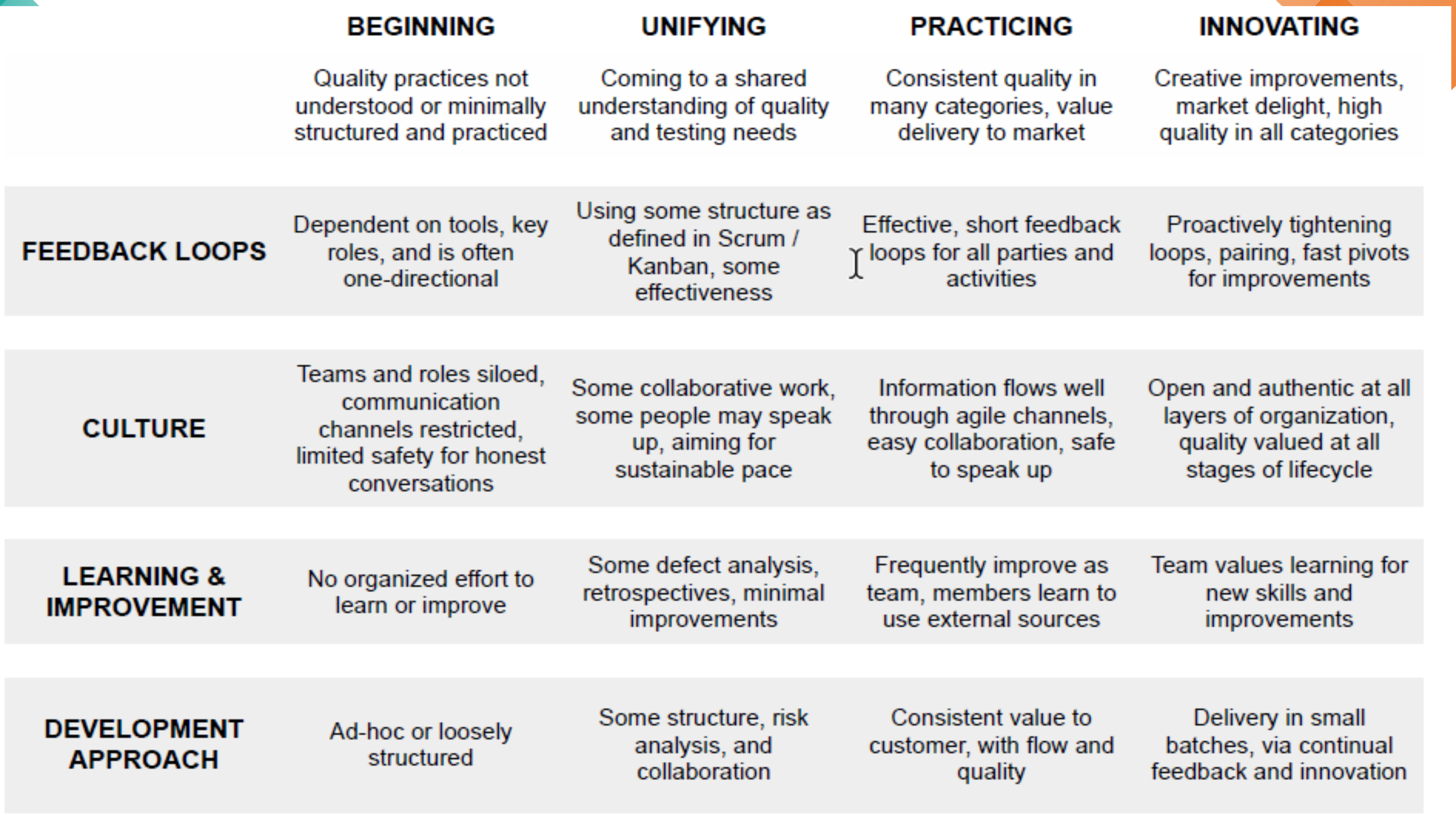
  Incorporate consistent feedback loops, write unit tests, and embrace frequent collaboration to eliminate problems early.

- **Development Approach**:

  Whole team will collaborate early to find user story problems, discover what the customer wants, and decide on designs and tests. Collaboration costs are cheaper than bug costs resulting from collaboration that didn’t happen.

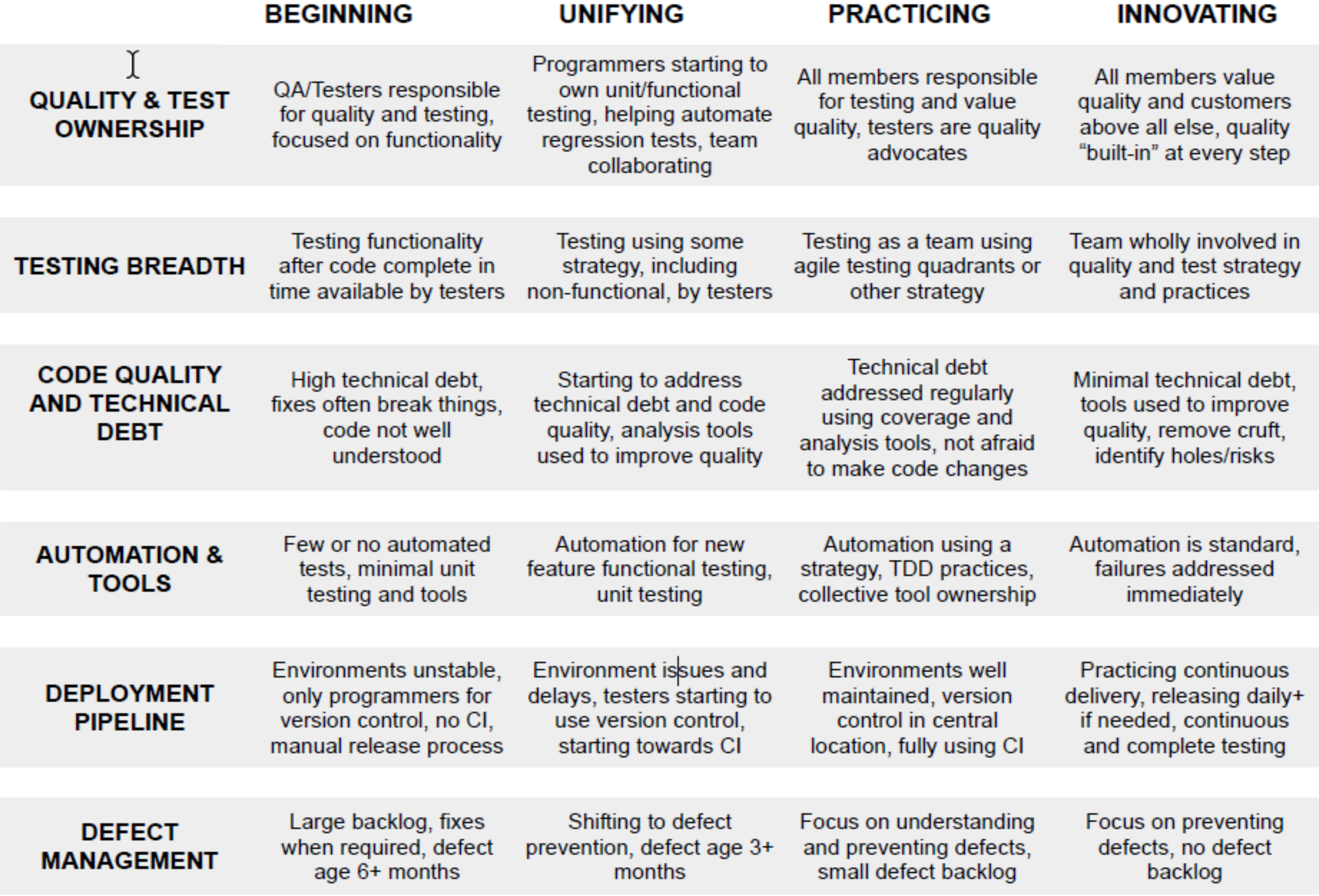
- **Code Quality**:

  Write more unit tests, for all new stories and features. They take less time to write than investigating and fixing a bug. Programmers would prefer to spend time coding new features.



	BEGINNING	UNIFYING	PRACTICING	INNOVATING
	Quality practices not understood or minimally structured and practiced	Coming to a shared understanding of quality and testing needs	Consistent quality in many categories, value delivery to market	Creative improvements, market delight, high quality in all categories
FEEDBACK LOOPS	Dependent on tools, key roles, and is often one-directional	Using some structure as defined in Scrum / Kanban, some effectiveness	Effective, short feedback loops for all parties and activities	Proactively tightening loops, pairing, fast pivots for improvements
CULTURE	Teams and roles siloed, communication channels restricted, limited safety for honest conversations	Some collaborative work, some people may speak up, aiming for sustainable pace	Information flows well through agile channels, easy collaboration, safe to speak up	Open and authentic at all layers of organization, quality valued at all stages of lifecycle
LEARNING & IMPROVEMENT	No organized effort to learn or improve	Some defect analysis, retrospectives, minimal improvements	Frequently improve as team, members learn to use external sources	Team values learning for new skills and improvements
DEVELOPMENT APPROACH	Ad-hoc or loosely structured	Some structure, risk analysis, and collaboration	Consistent value to customer, with flow and quality	Delivery in small batches, via continual feedback and innovation

@SelenaDelesie #Leadership #PNSQC



	BEGINNING	UNIFYING	PRACTICING	INNOVATING
QUALITY & TEST OWNERSHIP	QA/Testers responsible for quality and testing, focused on functionality	Programmers starting to own unit/functional testing, helping automate regression tests, team collaborating	All members responsible for testing and value quality, testers are quality advocates	All members value quality and customers above all else, quality "built-in" at every step
TESTING BREADTH	Testing functionality after code complete in time available by testers	Testing using some strategy, including non-functional, by testers	Testing as a team using agile testing quadrants or other strategy	Team wholly involved in quality and test strategy and practices
CODE QUALITY AND TECHNICAL DEBT	High technical debt, fixes often break things, code not well understood	Starting to address technical debt and code quality, analysis tools used to improve quality	Technical debt addressed regularly using coverage and analysis tools, not afraid to make code changes	Minimal technical debt, tools used to improve quality, remove cruft, identify holes/risks
AUTOMATION & TOOLS	Few or no automated tests, minimal unit testing and tools	Automation for new feature functional testing, unit testing	Automation using a strategy, TDD practices, collective tool ownership	Automation is standard, failures addressed immediately
DEPLOYMENT PIPELINE	Environments unstable, only programmers for version control, no CI, manual release process	Environment issues and delays, testers starting to use version control, starting towards CI	Environments well maintained, version control in central location, fully using CI	Practicing continuous delivery, releasing daily+ if needed, continuous and complete testing
DEFECT MANAGEMENT	Large backlog, fixes when required, defect age 6+ months	Shifting to defect prevention, defect age 3+ months	Focus on understanding and preventing defects, small defect backlog	Focus on preventing defects, no defect backlog

41ST ANNUAL PACIFIC NW SOFTWARE QUALITY CONFERENCE OCT. 9-11,2023

Amplifying Agile Quality Practices

Benefits of Using QPAM for a Quality Practices Assessment

- Consistent message for teams that is not "method" specific
- Teams know where they are… and can decide where they would like to improve
- Method agnostic (although meant for agile teams)

This is how teams AMPLIFY!

Remember, it is NOT a maturity model.

----- not all teams strive for innovation

What's Next?

Janet Gregory and I are writing a *Facilitator's Guide to* conducting team and organizational assessments. It includes examples and team case studies using the QPAM model.

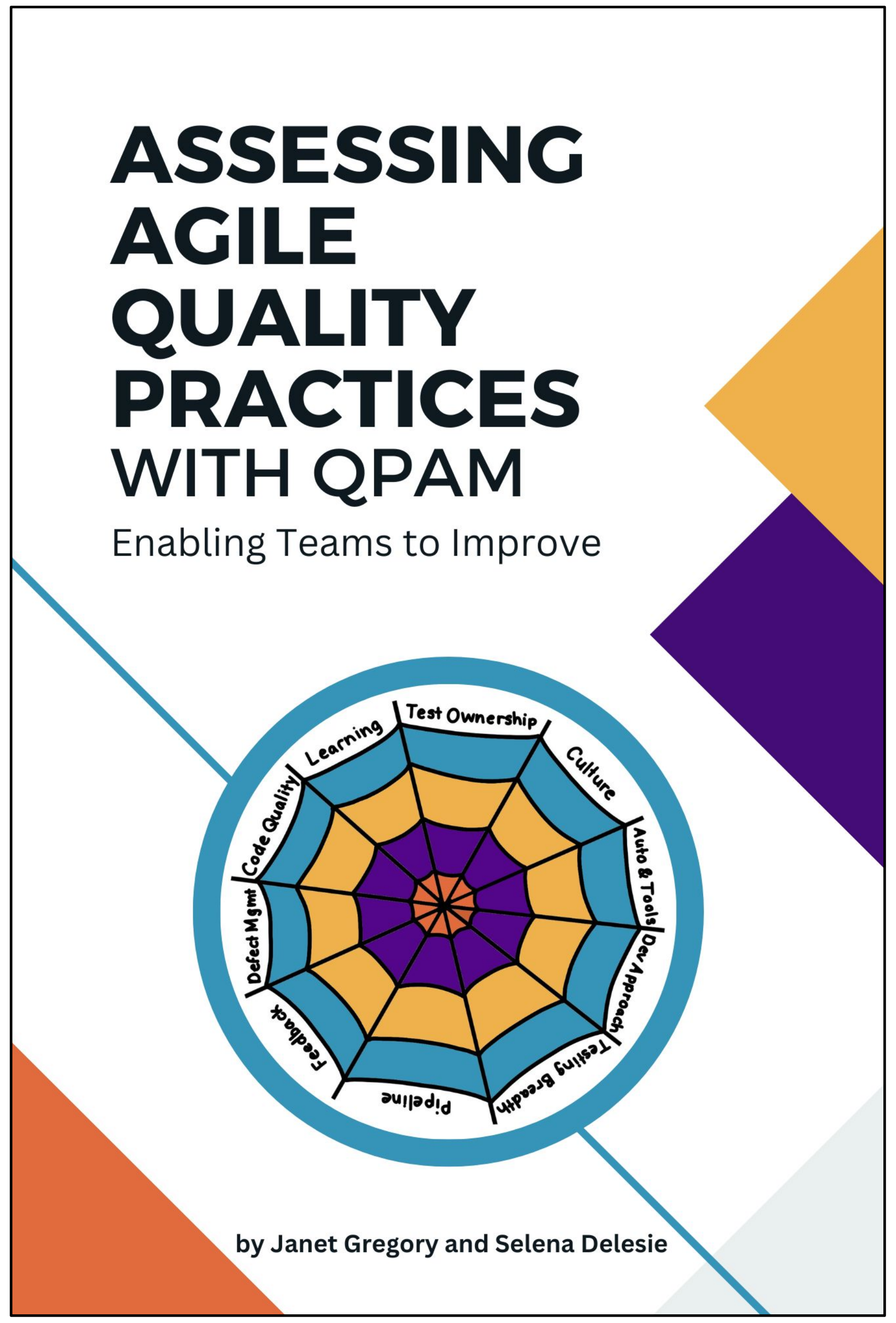
Book Release November 2023!

on LeanPub and Amazon

SCAN ME

Available on LeanPub

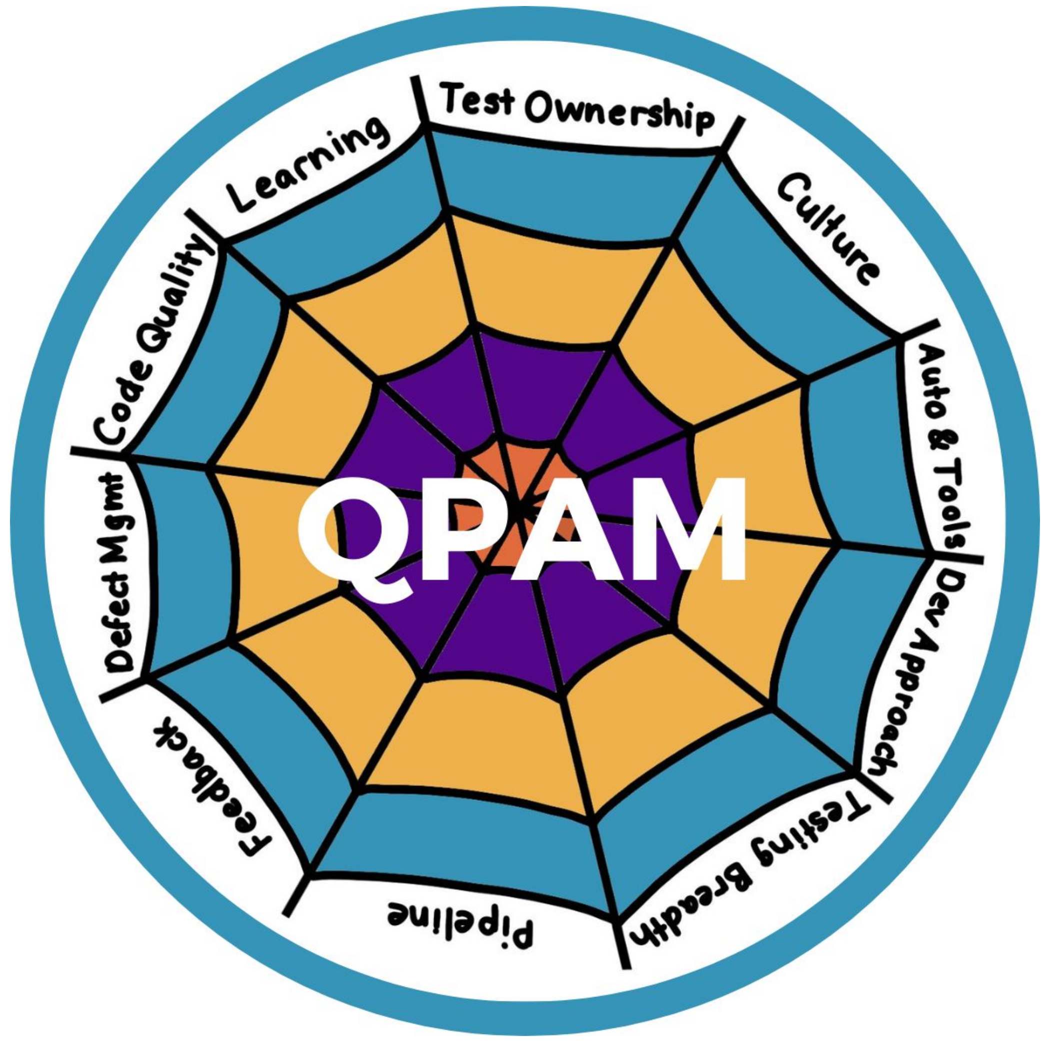
Amazon Release November 2023



ASSESSING AGILE QUALITY PRACTICES WITH QPAM

Enabling Teams to Improve

by Janet Gregory and Selena Delesie



PS. Find Me for…

QPAM Stickers

Brochure About the Book

SCAN ME

41ST ANNUAL PACIFIC NW SOFTWARE QUALITY CONFERENCE OCT. 9-11,2023