

SmartQA Wisdom



on

Mindset & Habits

Thiruvengadam Ashok

STAG Software

About the Book: SmartQA Wisdom on Mindset and Habits

SmartQA Wisdom on Mindset and Habits explores the essential mindsets and habits needed to build high-quality software and ensure effective testing. By drawing parallels between human well-being and software health, it highlights how maintaining structured practices leads to robust, maintainable, and adaptable code. Just as good habits contribute to physical and mental well-being, disciplined coding and testing practices create resilient and efficient software systems.

The book introduces the concept that **healthy code**, like a healthy body, requires consistent care and discipline. It outlines **seven habits for maintaining code health**, correlating software practices with human habits such as eating clean, staying active, and avoiding vices. It emphasises the importance of **future-proofing code**, ensuring maintainability, adaptability, and reliability—qualities that make software resilient and efficient.

Beyond technical health, the book delves into **mindsets for smart development and smart QA**, encouraging developers and testers to cultivate focus, clarity, and adaptability. It highlights the importance of **thinking beyond immediate tasks**, anticipating challenges, and designing solutions that integrate seamlessly within broader systems.

A key theme of the book is achieving a **state of flow**—a mental state of deep engagement where productivity and creativity flourish. By integrating structured processes with mindful immersion, developers and testers can elevate their performance, making their work more intuitive and efficient.

Blending practical methodologies with philosophical insights, *SmartQA Wisdom on Mindset and Habits* serves as a valuable resource for software professionals who seek to refine their craft, improve their problem-solving abilities, and build software that is not just functional but truly exceptional.

About the author

Thiruvengadam Ashok is the CEO of STAG Software Private Limited & Co-Founder of Pivotrics, based in Bengaluru, India. Ashok has dedicated his career to the pursuit of quality assurance in software, continuously evolving his approaches to match the needs of modern systems. He is the creator of HyBIST, an innovative approach to SmartQA that has revolutionised how teams approach hypothesis-driven testing.

Ashok's professional life is deeply intertwined with his personal philosophy. A passionate ultra-distance runner and long-distance cyclist, he applies the principles of endurance and exploration to his work, constantly seeking out new ways to improve software quality. He is also an avid wordsmith, using his love of language to weave both poetry and technical innovation into his life's work.

He holds an M.S. in Computer Science from the Illinois Institute of Technology, a Bachelor's degree in Electronics and Communication Engineering from the College of Engineering, Guindy, and a Postgraduate Diploma in Environmental Law from the National Law School of India University, Bangalore. His life maxim—"Love what you do & Do only what you love"—is reflected in everything he undertakes, from professional projects to personal passions.

Copyright © 2025, Thiruvengadam Ashok

All rights reserved.

No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations used in reviews and certain other noncommercial uses permitted by copyright law.

Disclaimer:

The information contained here is for educational and informational purposes only. While every effort has been made to ensure the accuracy of the content, the author and publisher make no representations or warranties regarding the completeness, accuracy, or applicability of the information provided. The strategies and methodologies described are for informational purposes and should be adapted to individual circumstances as necessary.

Trademarks:

All product names, logos, and brands mentioned in this book are the property of their respective owners. Use of these names, logos, and brands does not imply endorsement.

HyBIST is the intellectual property of STAG Software Private Limited.

Edition: First edition, 2025.

TABLE OF CONTENTS

Seven Habits for Healthy Code	6
Attributes of good health	6
For humans	6
For code	6
Habits and mindsets to satisfy these attributes	7
For humans	7
For code	7
Drawing parallels between human and code health	7
Role of mindset and habits in healthiness	8
Seven habits for healthy code	8
Mindset & Habits for Smart Development	9
Mindset for smart development	9
Habits for smart development	10
Mind map of mindsets & habits for Smart Dev	12
Mindset & Habits for Smart QA	13
Mindset for Smart QA	13
Habits for Smart QA	14
Mind map of mindsets & Habits for SmartQA	16
Be in Flow	17
The State of Flow	17
Achieving the State of Flow	17
The benefits of flow	18
In closing	18

SEVEN HABITS FOR HEALTHY CODE

Attributes of good health

For humans

Good health is not just the absence of illness; it is a combination of physical and mental well-being. When we think of being healthy, we associate it with several key attributes:

Fit	Toned and without unnecessary weight
Strong	Capable and efficient in daily activities
Glowing	Good appearance, reflecting inner health
Resilient	Resistant to infections and external threats
Fast Recovery	Quick to bounce back from illness
Inspiring	Influencing others to maintain good health

Achieving this state requires a balance between a healthy body and a healthy mind. It is not just about feeling good in the moment but also about ensuring long-term well-being.

For code

Just as a person needs to be fit and strong to function optimally, code needs to be structured in a way that ensures its longevity and effectiveness. Healthy code is not just about working correctly; it is about future-proofing, maintainability, adaptability, and reusability. Like a person's healthy appearance, well-structured and efficient code also shines.

The attributes of healthy code align with those of a healthy human:

Fit	Functional and testable
Strong	High performance and reliability
Glowing	Well-organised, readable, and communicative (in UI, text, API)
Resilient	Capable of handling errors and exceptions
Fast Recovery	Easily maintainable and supportable
Inspiring	Reusable and free from unintended side effects

Healthy code is not the result of mere reviews or testing. It is built through simple, consistent practices, just like a healthy body is maintained through good daily habits.

SMARTQA WISDOM ON MINDSET & HABITS

Habits and mindsets to satisfy these attributes

For humans

Health is the outcome of consistent habits. The following habits contribute to both a healthy mind and body:

- Eat clean
- Be active
- Avoid vices
- Rest well
- Practice hygiene
- Be calm
- Be flexible, adapt

These habits ensure the body stays fit, strong, and resilient while maintaining mental clarity and stability. They prevent long-term health issues and enhance overall well-being.

For code

The same principles that apply to maintaining human health can be applied to code. Healthy software is developed and maintained through consistent and mindful practices.

Eat clean	Keep your development environment clean
Be active	Visualise the flow
Avoid vices	Anticipate issues, avoid anti-patterns
Rest well	Write less, keep code concise
Practice hygiene	Use aids such as cheat sheets, checklists, guidelines, and patterns
Be calm	Abstract well
Be flexible, adapt	Refactor regularly

These habits ensure that code remains maintainable, scalable, and free from unnecessary complications.

Drawing parallels between human and code health

The relationship between human health and code health follows the same principles. A disciplined approach leads to long-term efficiency and reliability.

- Eating clean is like maintaining a clean and structured development environment
- Staying active corresponds to visualising how code flows and interacts
- Avoiding vices means anticipating problems and staying away from bad coding practices

SMARTQA WISDOM ON MINDSET & HABITS

- Resting well is akin to keeping code concise and avoiding unnecessary complexity
- Practicing hygiene involves using structured methods like guidelines and checklists
- Staying calm aligns with proper abstraction and code organisation
- Being flexible and adaptable corresponds to refactoring and continuous improvement

A well-maintained body and well-structured code both exhibit resilience, efficiency, and long-term sustainability.

Role of mindset and habits in healthiness

A positive mindset and healthy habits are fundamental to maintaining both human and code health. Health is not achieved by a single action but through sustained, mindful practices. In humans, a strong mindset ensures discipline in maintaining good habits. Similarly, in software development, a well-formed mindset ensures best practices are followed consistently.

Healthy code, like a healthy body, does not come from occasional fixes or last-minute interventions. It is built through constant refinement, structured approaches, and a focus on long-term efficiency. A proactive approach, rather than reactive fixes, results in code that is easier to maintain, more reliable, and adaptable to future needs.

Seven habits for healthy code

Maintaining code health requires a disciplined approach and adherence to best practices. The following seven habits are essential to achieving and sustaining high-quality code:

1. Abstract well
2. Visualise the flow
3. Refactor regularly
4. Write well (ensuring clarity and readability)
5. Use aids such as cheat sheets, checklists, guidelines, and patterns
6. Anticipate issues and avoid anti-patterns
7. Keep the development environment clean

Healthy code is not the result of reviews or testing alone. It is achieved by following simple, diligent practices consistently over time.

MINDSET & HABITS FOR SMART DEVELOPMENT

This explores essential mindsets and habits for smart development, enabling developers to craft exceptional, maintainable, and user-focused software. By embracing focus, simplicity, and adaptability, developers can approach coding challenges with clarity and purpose. Cultivating effective daily practices such as writing clean, testable code and refactoring regularly ensures long-term success. This encourages developers to remain inquisitive, empathetic, and mindful of the broader context in which their code operates. Together, these principles empower developers to create not just functional software, but elegant solutions that deliver true value.

Mindset for smart development

Adopt a mindset that shapes how you think, solve problems, and approach creativity in coding, reflecting core attitudes and beliefs to craft exceptional software.

1. Be focused

Focus on doing one thing at a time to maintain clarity and effectiveness.

In each code fragment, focus on solving one problem exceptionally well. Attempting to address multiple issues at once can lead to messy results. Stay single-minded in what your code is designed to solve.

2. Be defensive

Adopt a cautious and proactive attitude toward handling potential issues.

Accidents happen, and inputs may not always be clean or safe. Prepare for eventualities by adopting a defensive coding style. It's your responsibility to anticipate risks and safeguard your code.

3. Be expansive

Think beyond the immediate task to understand the broader context of your code.

Strive to grasp the larger context in which your code will be used. Deliver value by ensuring your code aligns with the bigger picture—"see the forest for the trees."

4. Be a good citizen

Develop a respectful and responsible attitude toward resource usage.

Respect the environment in which your code runs. Consume only the resources you need and release them promptly when they are no longer required. Avoid being wasteful.

5. Be efficient

Cultivate an awareness of performance, security, and other key factors in coding.

SMARTQA WISDOM ON MINDSET & HABITS

Writing code isn't just about achieving functionality; it's about addressing crucial aspects like security, reliability, compatibility, and performance. Be sensitive to these elements while coding.

6. Be beautiful

Treat your code as a form of craftsmanship and aim for aesthetic appeal.

Great code is not just about correctness—it should also exhibit a sense of beauty. Pay attention to the choice of names, structure, and organization to create code that is elegant and inspiring. Brilliant engineering can transform into art.

7. Never assume, ask, question

Adopt a curious and inquisitive mindset to refine requirements and clarify uncertainties.

Requirements are rarely complete; they evolve over time. Never assume when something is unclear—always ask questions and seek clarity to ensure precision in your work.

8. Think of behavior in terms of conditions

Focus on understanding and managing conditions to ensure compliant behavior.

Good behavior in code arises from adherence to conditions. Pay attention to all possible combinations of conditions and ensure they are well-managed to prevent errors.

9. Be friends with bug(s)

View bugs as learning opportunities to improve your skills and solutions.

Don't hate bugs; they teach you to code better. Embrace each bug as an opportunity to learn and improve, ensuring you find them before others do.

10. Strive to understand how your code will be consumed

Adopt a user-centric perspective to design code that integrates seamlessly with other systems.

Your code doesn't exist in isolation. Visualise how it will be used—by other developers, systems, or components—so you can anticipate future requirements and ensure robustness.

11. Be sensitive to NFRs

Prioritise non-functional requirements like performance, security, and scalability from the start.

Non-functional requirements (NFRs) like load, usability, and maintainability cannot simply be added later. Build sensitivity toward these aspects to create robust and future-proof code.

Habits for smart development

Cultivate daily practices and routines that ensure your code is clean, efficient, and maintainable. Focus on small, consistent actions that enhance your coding and lead to long-term success.

1. Be clean

Organise and structure your code to make it easier to read and maintain.

SMARTQA WISDOM ON MINDSET & HABITS

Reduce clutter in your code—it's not just about getting the work done somehow. Good organization and structure are essential for clarity and long-term readability.

2. **Be malleable**

Write flexible, adaptable code that can be easily modified or extended.

Avoid using hardcoded values or magic numbers. Keep your code pliable, allowing for easy modifications and extensions in the future.

3. **Be maintainable**

Develop habits that make your code easy to update and understand later.

Code often requires changes the moment it's executed. Write maintenance-friendly code, remembering that you (or someone else) may need to revisit it later. Over time, even you might forget why certain choices were made.

4. **Be testable**

Incorporate features that enable easy testing to ensure code reliability.

The ability to verify that your code behaves as intended is critical. Build in hooks and mechanisms that make it easier to test and validate your work, which is key to writing great code.

5. **Use smart checklists**

Develop a habit of using checklists to preemptively identify and prevent potential issues.

While coding, be mindful of potential pitfalls. Use checklists to sensitise yourself to what might go wrong, focusing on prevention rather than relying solely on testing to catch issues later.

6. **Treat code as a living entity**

Refactor regularly to keep your code clean, simple, and adaptable.

Code is never truly done—it evolves constantly. Refactor consistently to simplify and improve your code, understanding that time is your only constraint.

7. **Don't be scared to inject bad inputs**

Test for edge cases and incorrect inputs to identify vulnerabilities early.

Good input tests are necessary, but it's the bad inputs and unexpected settings that create technical debt. Tackle these early to ensure robustness and reduce future risks.

8. **Be purposeful of issues to find via unit tests**

Clarify and target specific issues at the appropriate testing level.

Different issues require different testing strategies. Be deliberate about what to address via unit tests versus higher levels of testing. Ensure clarity in your goals to make unit testing more effective.

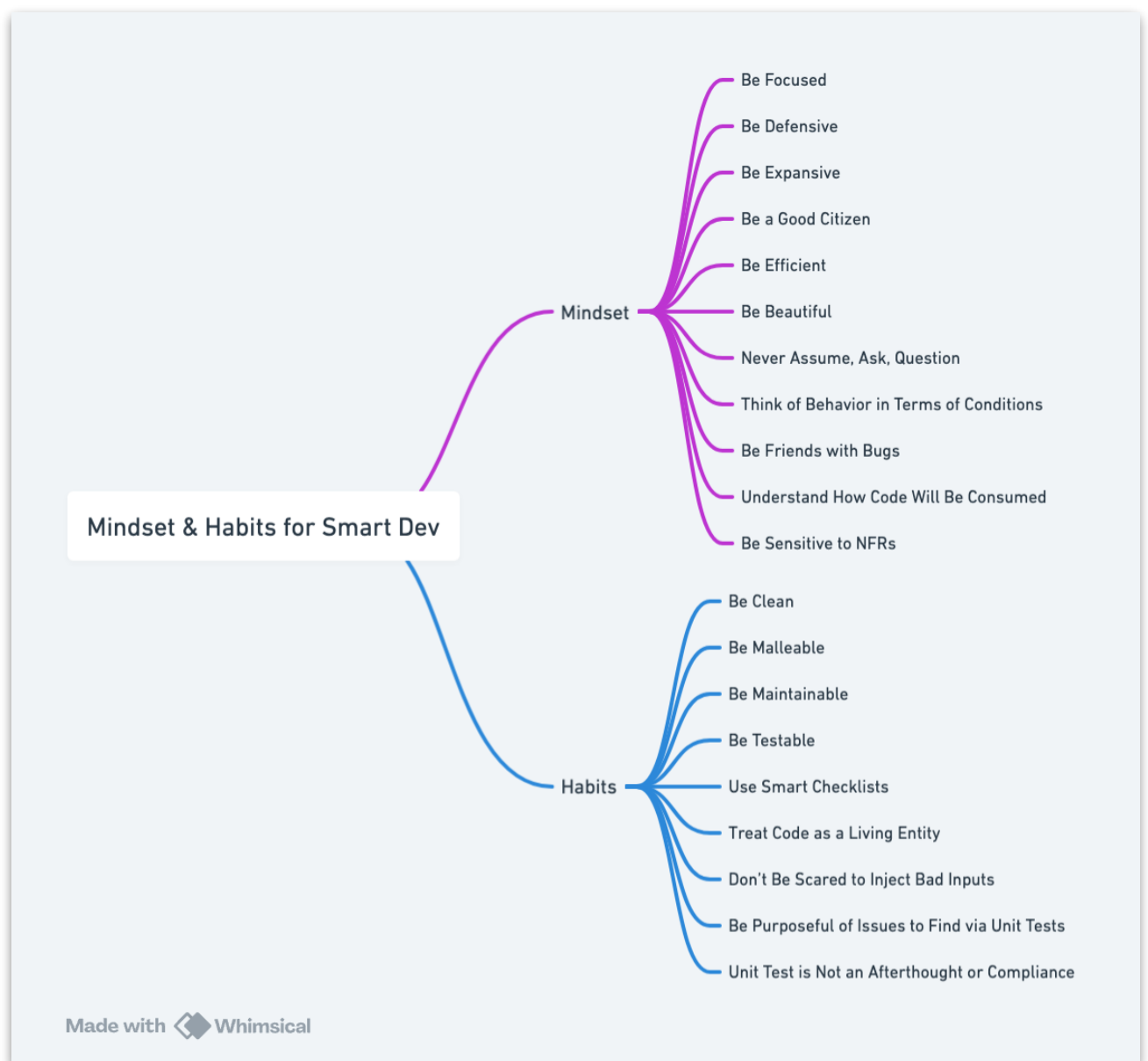
SMARTQA WISDOM ON MINDSET & HABITS

9. Unit test is not an afterthought or compliance

Incorporate unit testing naturally into your development process to ensure code reliability.

Unit testing is not a chore or a box to tick—it's an integral part of development. Write scripts or lightweight notes to simplify testing and make it repeatable. Treat testing as a seamless extension of coding to keep your workflow frictionless.

Mind map of mindsets & habits for Smart Dev



MINDSET & HABITS FOR SMART QA

This delves into the essential mindsets and habits every tester should cultivate to excel in their craft. It emphasises the importance of focus, simplicity, and empathy in approaching testing challenges while staying adaptable and inquisitive. Testers are encouraged to adopt purposeful practices, continuously refine their processes, and embrace change for improvement. By leveraging tools, visualising the bigger picture, and grounding decisions in facts, testers can enhance their efficiency and impact. Together, these mindsets and habits empower testers to deliver meaningful value through their work.

Mindset for Smart QA

A mindset defines how you think, analyze, and approach challenges in coding and testing with purpose and clarity.

1. Focus... Focus...

Train your mind to concentrate on what truly matters and eliminate distractions.

Focus on the who, what, where, and the purpose behind your actions. Prioritise value and relentlessly discard noise and unnecessary elements.

2. Analyse... Analyse...

Adopt a habit of questioning and evaluating to optimise efforts.

Should I regress this? Should I test in this environment? Should I document so much? Constantly analyze what you are doing and explore ways to do less while achieving more.

3. Sharpen...

Refine your goals and objectives to maximise impact.

Stay purposeful in your actions. Understand the issues that matter, their business benefits, user experience, and potential impact. Sharpen your objectives, set clear routes, and stay focused on them.

4. Simplify... Simplify...

Embrace simplicity in thought and execution.

If something is too complex to understand, break it down. If it's too complicated to execute, decompose it. Relentlessly simplify and ensure clarity is never compromised.

5. Empathise.

Understand and connect with others to deliver meaningful value.

SMARTQA WISDOM ON MINDSET & HABITS

Step into the users' shoes to grasp their expectations, dislikes, constraints, and the value they seek. Testing isn't just technical—it's about delivering value to users effectively.

6. **Visualise well, see the big picture of the system.**

Think holistically to understand how systems work and interact.

Visualise who uses the system, where and when they use it, and through which interfaces. Consider the architecture, control flow, dependencies, and technologies to appreciate the bigger picture.

7. **Unlearn.**

Be willing to let go of outdated knowledge to grow and adapt.

Revisiting what you know is as important as learning new things. Understanding is not about accumulation—it's about refining by discarding the irrelevant.

Habits for Smart QA

Habits are consistent practices that enhance your efficiency, organization, and adaptability in coding and testing.

1. **Practice... Practice... Practice...**

Engage in repeated actions to build expertise and speed.

Practice exploration, looking for bugs, modelling behavior, writing tersely, scripting, and observation. Through consistent practice, develop the ability to work unconsciously, speedily, and well.

2. **Steer... Constantly steer...**

Develop the habit of regular adjustments and improvements.

Course correct, adjust, revise, improvise, adapt, and repeat. As you iterate, refine your skills and approaches to become quicker and better.

3. **Immerse...**

Be fully present and mindful in your actions.

Stay aware of what you are doing now. Immerse yourself in the present moment, allowing yourself to explore deeply and centre your focus to achieve more.

4. **Discard...**

Learn to let go of what doesn't work or add value.

When something isn't working, discard it. If you discover you're wrong, discard the old approach. Metamorphose continuously by discarding and evolving.

SMARTQA WISDOM ON MINDSET & HABITS

5. **Organise...**

Bring structure and discipline to your work.

Set goals, plan, execute, and observe. Stay organised in your mind, as discipline and structure help smooth disruptions and chaos.

6. **Leverage...**

Use available resources, tools, and patterns to improve efficiency.

Recognise patterns and exploit them. Learn from others' work, use smart checklists, tools, and reusable strategies, scenarios, and scripts to save effort and boost productivity.

7. **Prove with facts.**

Ground your findings and decisions in evidence.

Show proof of test adequacy, hypothesise potential issues, and demonstrate how they can be detected. Prove why specific regressions are necessary to avoid unnecessary work.

8. **Iterate continuously.**

Adopt an experimental approach to refine your understanding.

Testing is about experimenting and exploring. Iterate with small changes to understand better and observe more effectively, constantly chewing on details to digest insights.

9. **Simplify the problem-solving.**

Ensure solutions address problems directly and effectively.

End users rely on systems to solve problems. Great software addresses their needs with simplicity. Good testing ensures systems do just enough to solve problems effectively without overcomplicating.

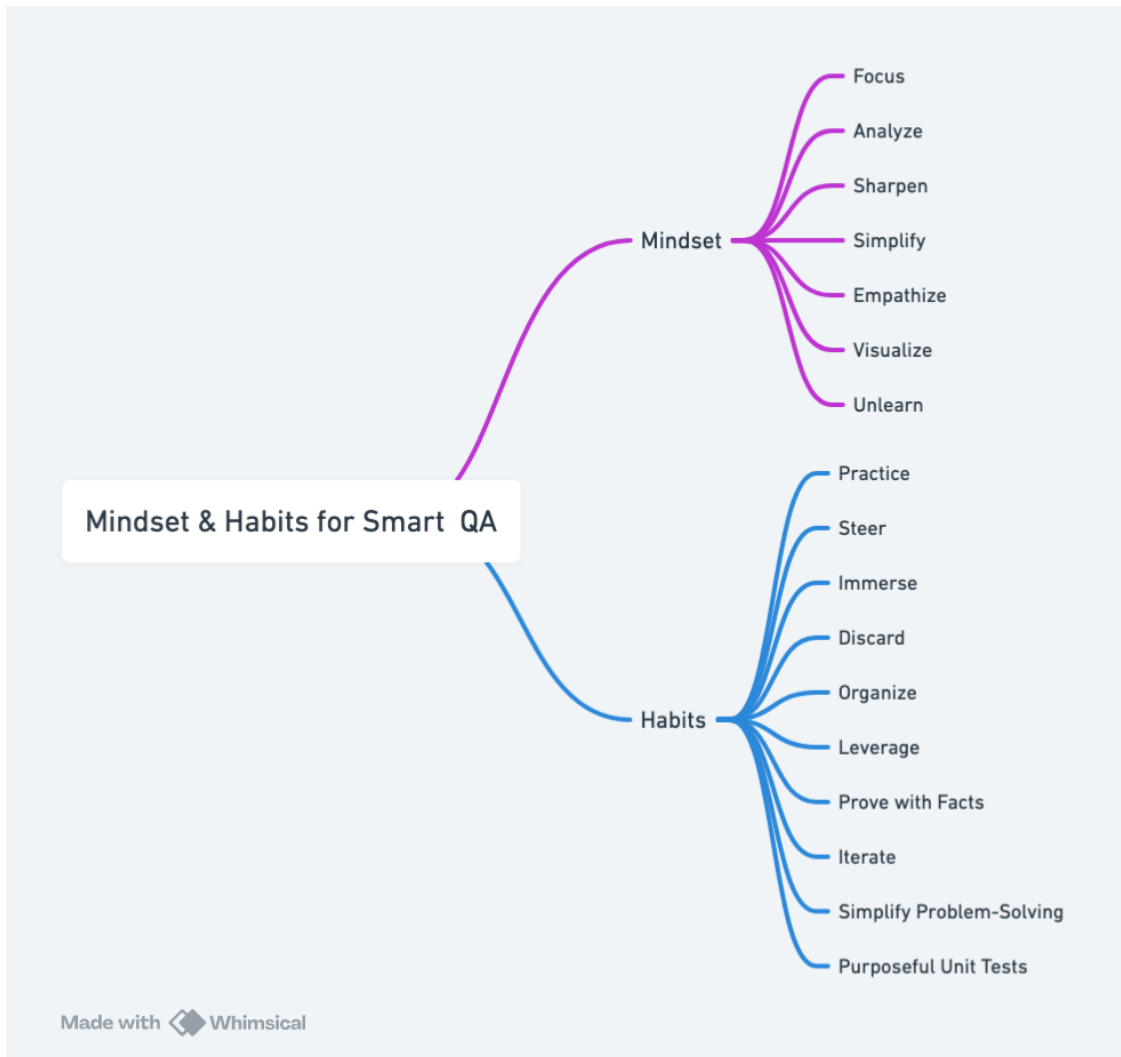
10. **Be purposeful of issues to find via unit tests.**

Be deliberate in your testing efforts to maximise their value.

Understand the types of issues to prevent, target via unit tests, or address at higher levels of testing. Maintain clarity in your testing goals to ensure focus and efficiency.

SMARTQA WISDOM ON MINDSET & HABITS

Mind map of mindsets & Habits for SmartQA



BE IN FLOW

Good testing is a seamless blend of intellect, techniques, processes, and heuristics. What does it take to achieve brilliance in testing? It requires complete immersion in the act, maintaining focus while remaining open, being keenly observant yet non-judgmental, and producing outcomes that are both effective and elegant. When testing is performed with such intensity and engagement, the experience becomes so fulfilling that time seems to stand still—a true state of flow. What does it take to reach this state?

Engineers rely on techniques, tools, processes, and intellect to achieve excellence. However, to go beyond good testing and achieve brilliance, it becomes essential to look beyond external resources. Having exhausted all external means—tools, techniques, processes, and intellectual effort—it is time to explore the vast internal potential that every individual possesses.

This requires moving beyond the conscious intellect, tapping into the subconscious mind, and harnessing its boundless capabilities. A practical step toward achieving this is to enter a state of flow.

The State of Flow

Flow is a state of complete immersion, where the mind is fully engaged in the task at hand. In this state, energy is channelled effortlessly, allowing work to be performed with agility and precision. It is a state where actions feel natural, requiring no forced effort, and time appears to slow down. In flow, efficiency increases, work feels effortless, and absolute joy is experienced.

Achieving the State of Flow

Reaching a state of flow is facilitated when multiple sensory inputs harmonise. This involves engaging various senses effectively—visual, tactile, and auditory stimuli working together to enhance focus and immersion.

- **Visual stimulation:** The use of colour, images, mind maps, and structured visual text aids comprehension and engagement. Mind maps, diagrams, and sketch notes can help in understanding the system under test, designing test strategies, recording test ideas, noting observations, listing scenarios, and tracking issues.
- **Tactile engagement:** Writing with a pen or stylus on paper or a tablet provides a sense of freedom that a keyboard may not offer. The physical act of writing or drawing enhances thought processes and allows ideas to flow naturally.

SMARTQA WISDOM ON MINDSET & HABITS

- **Auditory engagement:** Engaging the auditory senses by speaking aloud, whistling softly, or playing gentle background music can enhance focus. Those who respond well to sound may find that a suitable melody aids in achieving flow.

Achieving peak performance in a state of flow also requires working in structured sessions. Testing is best conducted in focused time blocks, typically lasting between 45 to 90 minutes. The key is to establish a clear objective at the beginning of each session and immerse fully, using sensory engagement techniques to enter a deep state of concentration.

The benefits of flow

When fully engaged in a state of flow, thinking is no longer limited to left-brain logical deduction or right-brain creativity. Instead, it becomes an expansive, multidimensional process driven by harmonised sensory inputs. This heightened state enables keen observation, rapid absorption of information, and deeper insights.

In this state, the subconscious mind, which is significantly more powerful than the conscious mind, becomes an active participant. Unconventional yet valuable questions arise, innovative ideas emerge quickly, and the smallest observations are captured—leading to highly effective and insightful testing.

In closing

Test automation increases efficiency, and machine intelligence enhances accuracy. However, to complement rather than compete with machines, testers must go deeper—tapping into their subconscious to elevate their capabilities. By entering a state of flow, testers can unlock their true potential and achieve brilliance in testing.



"We are SmartQA evangelists. For over two decades we have transformed how individuals, teams and organisations have practised testing. We espouse methodology to test intelligently. Our mission - Elevate to high performance via SmartQA."
www.stagsoftware.com



The HyBIST Approach to SmartQA - MASTERCLASS

Testing is deep probing to seek clarity and in the process uncover, preempt issues rapidly. The HyBIST approach enables designing smart probes and probing the system smartly.

<https://smartqa.academy/courses/smartqa-using-hybist>



doSmartQA - AI based Smart Probing Assistant to interrogate, hypothesise issues, design & evaluate user story or a set of stories in a sprint rapidly. An assistant for smart session-based testing based on HyBIST.

Download personal edition from [here](#)



SmartQA Musings - A gentle flurry of interesting thoughts on smart assurance as a weekly webcast. A refreshing view of assurance to broaden & deepen thoughts/actions.

www.stagsoftware.com/subscribe

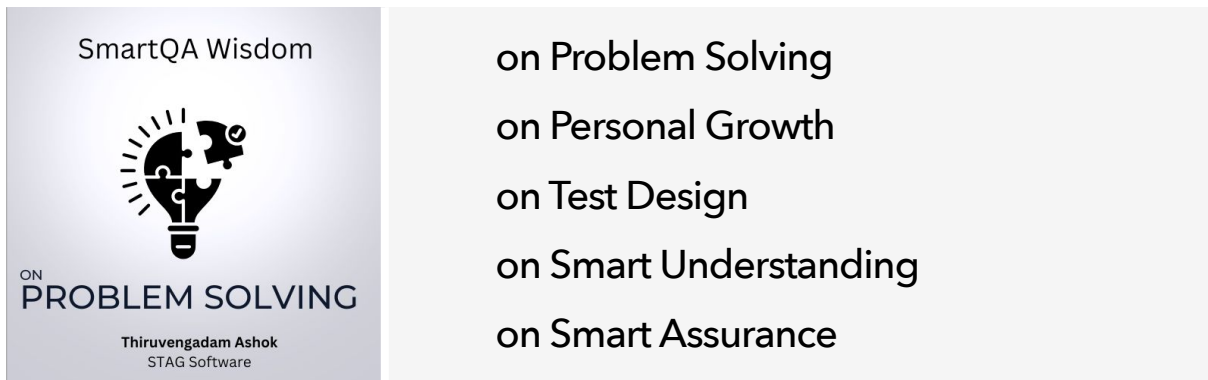


SmartQA Biweekly - Ignite your curiosity with fresh insights, thought-provoking ideas, and inspiring content delivered straight to your inbox every fortnight.

www.stagsoftware.com/subscribe

A rich collection of original content on smart assurance

SmartQA Wisdom - Profound nuggets of wisdom to think deeply, do rapidly & smartly for Test Practitioners.



Download from www.stagsoftware.com/smartqa-wisdom

SmartQA eBooks - for Sr Engg/QA managers, Sr Test Practitioner & Young Test Practitioners too.

HyBIST at a glance

do SmartQA -The HyBIST Approach

High Performance QA

50 Tips for SmartQA

Communicate Clearly



Download from www.stagsoftware.com/smartqa-ebooks