

How to become a Software Architect?

Vidya Vrat Agarwal

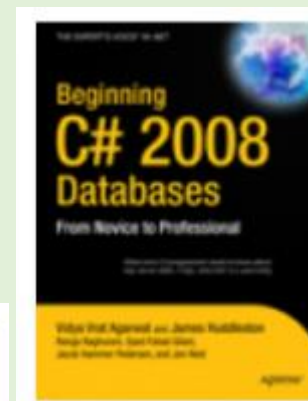
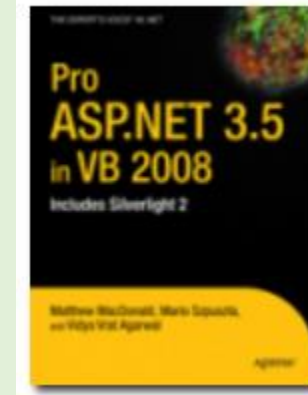
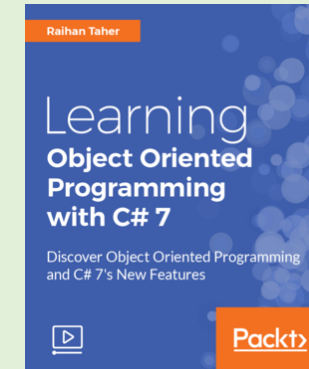
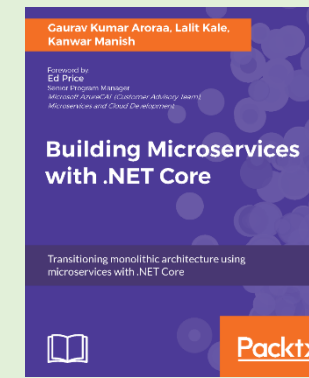
Principal Architect T-Mobile Inc.

www.MyPassionFor.Net | [@dotNetAuthor](https://twitter.com/dotNetAuthor)

<https://www.linkedin.com/in/vidyavrat/>

About Me

- Over 18+ years of industry experience
- Working as Principal Architect with T-Mobile Inc USA
- Microsoft MVP
- C# Corner MVP
- TOGAF Certified Architect
- Certified Scrum Master (CSM)
- Microsoft Certified (MCT, MCSD / MCAD .NET, MCTS etc.)
- Published Author (5) and Technical Reviewer (over a dozen)



Software Architect


← → ↻ Secure | https://www.google.com/search?q=software+architect&rlz=1C1GCEB_enUS757US757&oq=software

Google software architect 🔍

All Images Videos News Shopping More Settings Tools

About 174,000,000 results (0.59 seconds)

A **software architect** is a **software** expert who makes high-level design choices and dictates technical standards, including **software** coding standards, tools, and platforms. The leading expert is referred to as the chief **architect**. (Oct 1, 2017



lifehacker.com

The Path to Becoming a Software Architect – Nikolay Ashanin – Medium
<https://medium.com/.../the-path-to-becoming-a-software-architect-de53f1cb310a>

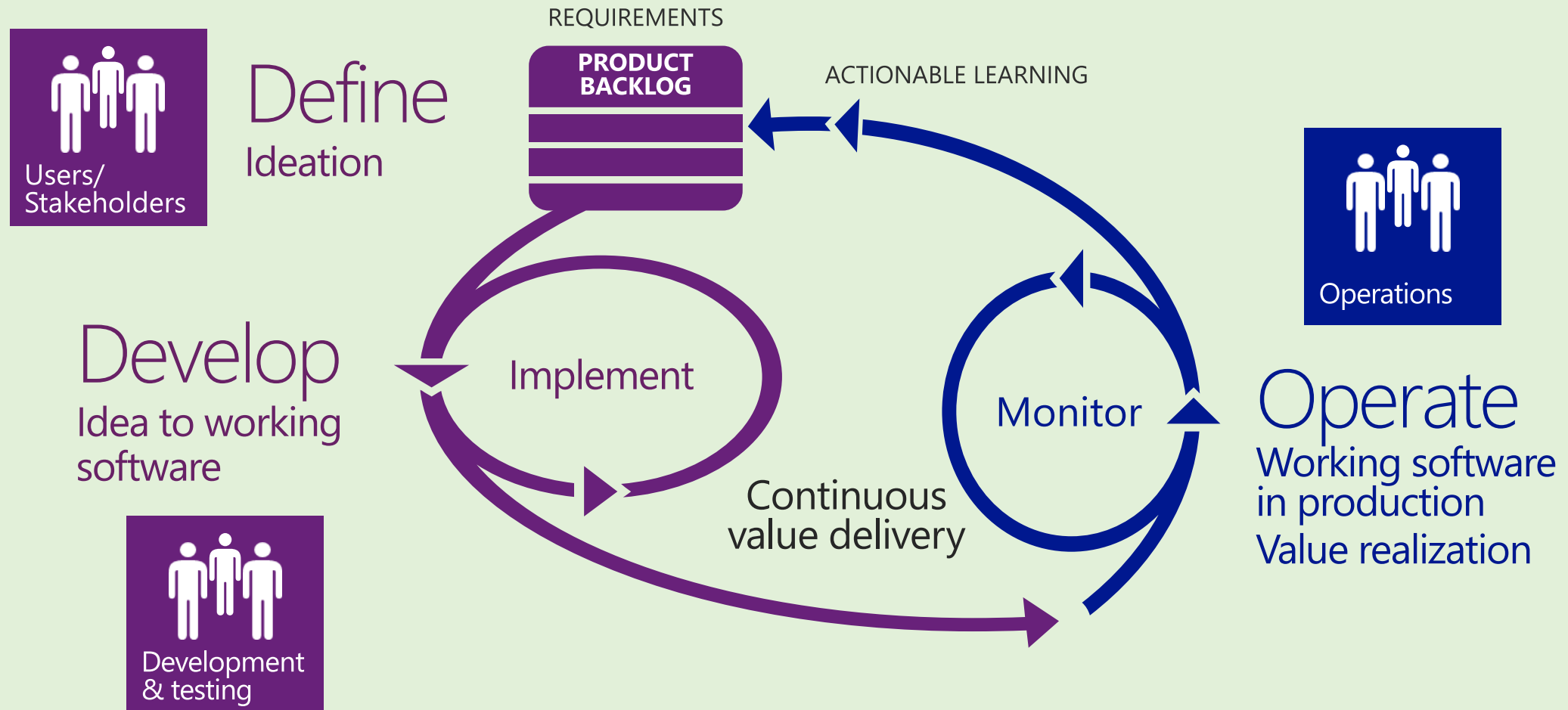
Is this something you want to be?

Why?

Core qualities a Software Architect must have

- Have technical background
 - Must have been through complete SDLC of various types of applications (Win, Web, Mobile etc..)
 - Must understand core concepts of technology
 - Hands on coding is a must
-
- Communication & Collaboration
 - Mentoring
 - Presentation
 - Drive results
 - Dive Deep
 - Big Picture
 - Ownership & Accountability
 - Backbone & High standards
 - Influential

Have full hands-on view of SDLC



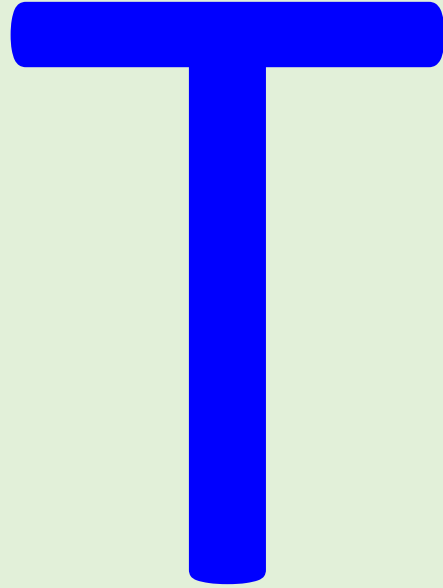
Certifications

- May not bring working knowledge
 - But certainly brings credibility
 - Combination is deadly.
-
- I have over 10 MS certifications
 - TOGAF
 - CSM
 - CAPM

Architecture is all about Patterns.

Do Architects know everything?

What is the Shape of an Architect?



How does a day look like?

Not in a specific/priority order

1. Meetings
2. Roadmap / discussions
3. Impact analysis
4. Design discussions
5. Production issues
6. Performance improvements
7. Code quality / Best Practices
8. Dashboards
9. Process improvements
10. Cross-team-collaboration
11. Hands-on-Code
12. Code Reviews
13. Mentoring/coaching
14. Innovation
15. Leadership discussions

Start Small, Think Big

1. Get the context of the problem.
2. Ask clarifying questions/understand requirements
3. Cut the ambiguity
4. Think of customer journey
5. Think long-term
6. Design a reactive system
7. Own the design and handoff to TA/BA/Dev etc.
8. Validate

Fake it until you make it.

Steven Tyler



Thank You

Vidya Vrat Agarwal

www.MyPassionFor.Net | [@dotNetAuthor](#)

<https://www.linkedin.com/in/vidyavrat/>