Survival Learning Skills For the Software Developer

By Brian Peterson



Survival (Learning) Skills

You have just joined a team that is using technologies that are new or unfamiliar.

The feeling of being lost in the wilderness comes over you and soon you've gone into survival (learning) mode.

How do you provide value to the team, keep your sanity, and learn what is necessary to excel.



Who Am I?

- Brian Peterson
 - Technical Director with nvisia.
 - 27+ years in Software Development.
 - 19+ year with nvisia.
 - Enjoy Enterprise and System Architecture
 - Mentoring and solving difficult problems
 - Java, .NET, Containers, Azure, and many more.
 - Former BSA Asst. Scoutmaster.







A new project is like being lost in the wilderness. There are many challenges and opportunities. The well-prepared will succeed.



Why is it important to have survival skills?

- New company or a new project can be highly stressful.
- You only make one first impression.
- The number of knowns and unknowns can be lopsided.
- The expectation that you come up to speed quickly and contribute.
- The technology stack may be completely new to you.
- Projects can change quickly and without notice.



Most Important Survival Skills

Shelter Building

Observation Skills

Fire-starting

Navigation and Orientation

Finding Food and Water

Self Defense

Basic First Aid

Outdoor Cooking

Advanced Knot-Tying



Shelter Building

A place of shelter is important to be safe and grow skills.

- Find a Mentor.
- Create a sandbox to experiment.
 - Containerization can by your friend.
- Admit what you don't know.
 - Opportunities for growth.
 - Leverage Mentor to fill gap.
 - Seek training.



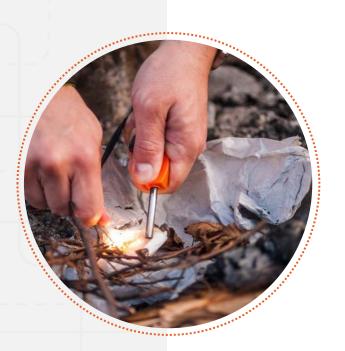


Observation Skills

- When joining a new project, look at how the team operates.
- What is the general mood of the team?
 - Who are the leaders? Who are the followers?
 - Is the perceived quality of the product good or bad?
- What is the approach for Code quality and review







Fire Starting

- Determine your motivation for this project (light your fire)
 - I want more tools in my belt (+)
 - Exciting technology (+)
 - Improve the product (+)
 - Show what I know (+)
 - I want to be an Expert in something (+)
 - Part of the job/it needs to be done (-)
 - Finish as quickly as possible (-)



Several parts to Food and Water

- Gather food
- Clean water
- Hunting with primitive tools





- Identification, foraging, and preparation
 - Read the Manual / Team Documentation
 - StackOverflow vs. ChatGPT vs. Source of Truth
 - Collect good sources of information:
 - Books, Blogs, Papers
 - Keep a notebook
 - Design Patterns used
 - Architecture Scenarios
 - Challenges and Outcomes





- Water Filtration and Purification
 - Watch out for stale information
 - Test what you use
 - Validate something is safe.
 - Don't trust everything you read on the Internet.
 - Are you legally able to use it?
 - GPL, LGPL, BSD, MIT, etc.





- Primitive Weaponry
 - What is behind the target?
 - DevTest deployment vs Production deployment.
 - Related features?
 - Demo?
 - What is your target?
 - The "POC to Production" range
 - What is the timeline?





Navigation and Orientation

- Define small goals and check your progress.
- Avoid going in circles.
 - Ask for help if you get lost.
 - Timebox your investigation/self-help
- Drop breadcrumbs
 - For difficult problems, as you are trying to figure something out, commit locally and with solid commit comment
 - Use Comments that speak to <u>Why</u> you are doing something... not <u>What</u> you are trying to do.
 - Add **TODO** comments to mark things you need to come back to.
 - IDEs often make it easy to find those waypoints.
 - Leave No Trace clean up when done





Self Defense

- Assume challenges are of positive intent.
 - If clearly negative intent, bring allies.
- Testing is part of "done"
 - Write unit tests that validate specifics.
 - Automate those tests so they don't rot.
- Frequent validations of working things
 - Keep code reviews small
 - Consider validating with your mentor or trusted team member.
- If you copy/paste something, know how to defend why it is correct.
 - Don't just say "StackOverflow had it as the accepted answer."





Basic First Aid

- You will make mistakes.
 - Take notes so you don't repeat them.
 - Write more tests to avoid the mistake
- Leverage Static Analysis tools to reduce impact.
 - Use the tools regularily.
- Beware of festering code.
 - Don't make it worse. If it is yours, clean it up right away
 - Plan for replacement if possible or at least document the risk with the code.





Outdoor Cooking

Everyone has basic needs, and this is true when joining a new company or project.

- Get outside and take a walk
- Take time to eat while learning
 - Fire up the grill and add family/friends
- Get proper sleep
- Maintain a healthy work/life balance.
 - Friends and Family are more important at the end of the day.





Advanced Knot-Tying

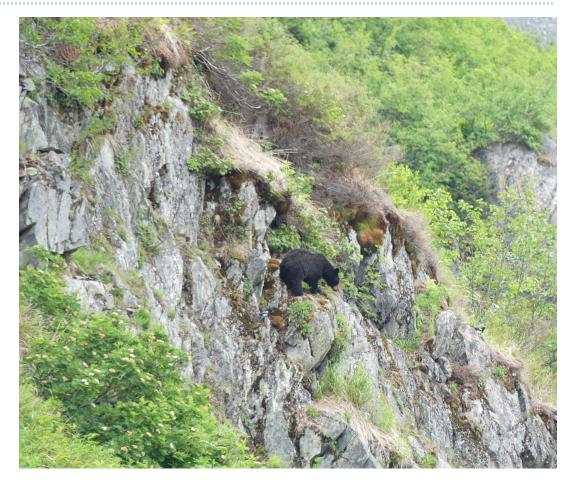
- Use the right thing for the context
- Make sure that it is secure
- Know the weakness of your context





Wilderness and Software Survival

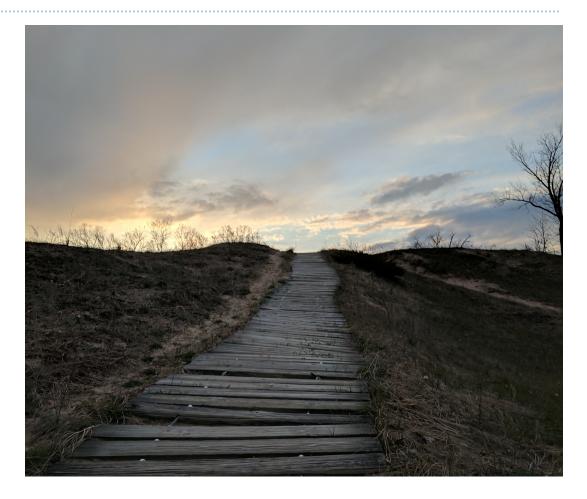
- Read the Manual
- Test what you write
- Have a mentor or two
- Learn something new
- Know when to ask for help
- Don't trust everything you read on the internet
- Make sure to look around





Happy Trials Trails

Get out with Friends and FamilyEnjoy the journey





Brian Peterson

Email:bpeterson@nvisia.comLinkedin:brian-peterson-3bb6664

