ROLE DESCRIPTION

SYSTEMS ENGINEER

WILDBIT

crewmojo

WELCOME TO

CREWMOJO

Enabling performance experiences to be designed and delivered in days not months.

Use these templates for inspiration or as a starting point for your own system. When you need to automate and scale the process, each template is ready to go in the Crewmojo platform.

Take a personal tour of Crewmojo.

Systems Engineer

This document describes the role by focusing on good and bad behaviours to clearly indicate what success looks like. It has been open-sourced from the Wildbit Good Bad Project

Role:	Systems Engineer
Level:	Individual Contributor
Skills & Behaviours	Curious Learners
	Data Driven
	Strong Writer
	Communicator
	Improves processes
	Creates Psychological Safety
	Systems Thinker
	Solves Customer Problems
	Flexible
	Prioritises Effectively
	Resource Aware
	Outcome Focused
Credit	Wildbit Good Bad Project

Curious Learners

A good systems engineer is addicted to learning, expanding their skill set, and improving their craft. They are excited about working with a breadth of technologies.

A bad systems engineer wants to take shortcuts and engages in half-assery. They do not take the time to document things because it's “hard.” They don’t take the time to automate because it's "complex.” They are not personable or respectful of other roles and don't take the time to communicate with the rest of the team because it's uncomfortable.

A good systems engineer can admit when they are weak in an area and has the motivation to strengthen themselves in this area instead. In other words, they don't skip leg day.

Data Driven

A good systems engineer doesn't ignore experience or intuition, but still uses 99% empirical evidence, 1% intuition to make decisions.

A bad systems engineer uses 99% intuition or opinion and 1% empirical evidence to make decisions.

Strong Writer

A good systems engineer not only writes thoughtfully-organized documentation but enjoys writing documentation and considers their audience when doing so.

A bad systems engineer postpones writing documentation until the last minute, and doesn't think about their audience they're writing for, or says things like, "My code is self-documenting.”

A good systems engineer writes useful, pertinent, and accessible documentation. A bad systems engineer copy and pastes a man page.

Communicator

A good systems engineer explains concepts to a non-technical audience in a way their audience can understand and moreover, appreciate. Instead of explaining the technical details they find valuable they explain the useful takeaways they know their audience will find valuable.

Improves Processes

A good systems engineer always thinks about unnecessarily repeated processes and how they can be automated, and do so in a way that is clear and easily maintainable for when these processes inevitably require iteration or improvements.

A bad systems engineer doesn't consider automating repeated processes and performs common tasks manually.

Creates Psychological Safety

Human beings make mistakes, and it’s amateurish to point fingers or harp on these mistakes. A bad systems engineer will do this regularly.

A good systems engineer will quickly fail and move on, and when mistakes are made by other members of the team, will talk to them as an equal in private, with the goal of acknowledging and moving past the mistake as quickly as possible. A good systems engineer tries to accentuate the positive and eliminate the negative.

Systems thinker

A good systems engineer considers the holistic implications of changes they make to a system and communicates with anyone who may be affected by these changes. A good systems engineer demonstrates an attention to detail.

A bad systems engineer doesn't consider anything but the small scope of what they're working on. They rush to solve a problem but instead create a new one because they didn’t stop to think.

A good systems engineer anticipates the needs of their team. They consider themselves a partner to developers and customer success, not a resource. Therefore, they do not work in silos.

Solves Customer Problems

A good systems engineer stays calm but earnest under pressure. They tackle a problem with a clear head, and an acute understanding of the impact the issue is having on the user.

A bad systems engineer resolves issues haphazardly. They don’t spend the extra minute to understand the entire situation and therefore end up running around in circles around the actual issue. .

Flexible

A good systems engineer has strong opinions, loosely held. This means a good systems engineer understands what makes a technology desirable, and they are honest about its shortcomings. They wholeheartedly consider multiple options and aren't bound by ideology. A good systems engineer takes a casual, inclusive, open approach in conversations with other team members.

A bad systems engineer is stubborn -- they have strong opinions and want everyone else to adhere to their opinions. A bad systems engineer takes on an adversarial, exclusive, or self-righteous tone in conversations with other team members.

Prioritises Effectively

A good systems engineer doesn't prioritize work based on how technically complex or interesting it will be to work on, but how useful it will be to their team or customers.

A bad systems engineer works on projects that are interesting to them, without considering the benefit it will provide to others or even vetting it and collecting thoughts from others first.

Resource Aware

A good systems engineer is frugal but considers time/cost of solving a problem versus the rare need to purchase a solution in an open source environment. Anything that can save time in working toward a solution also inherently saves money and therefore is financially conscientious. They also consider when it's appropriate to ask for support from other team members or external support channels.

A bad systems engineer immediately wants to purchase out-of-the-box solutions for problems and doesn't communicate or take part in the open source community for support or education.

Outcome Focused

A good systems engineer solves the problems at hand.

A bad systems engineer engages in over-engineering. They tend to inadvertently create new problems while solving old ones, or they spend their time solving problems nobody will ever realistically encounter.

EXPLORE MORE

Building a world-class performance culture is made easy with our template library and pre-designed employee experiences.

Templates:

- One-on-one templates
- Performance review templates
- Role descriptions
- Goal templates
- Survey templates
- Engagement surveys
- and more

Experiences:

- Onboarding new employees
- Goal setting & alignment
- Growth plans & coaching
- Skill tracking & development
- Feedback & recognition
- Stay interviews
- Performance reviews
- Exit surveys & interviews
- and more

View Templates

Book a Demo