

TECHNICAL ENGINEERING DUE DILIGENCE CHECKLIST

Try Smartsheet for FREE

SYSTEM ARCHITECTURE	
	What's the overall architecture of the system?
	Is it well-documented?
	Is it easily understood?
	Does it use industry standard components?
	Are there many vendors (e.g. AWS, Azure, etc.)?
	What's the next architectural change?

PERFORMANCE MONITORING	
	What monitoring is happening now?
	Has load testing ever been performed?
	What are the potential bottlenecks?
	What remedies are in the works for potential bottlenecks?

SCALABILITY	
	Does the system use load balancing?
	What are the separation of functions?
	Are there any single points of failure?
	Does scaling happen automatically?
	What isn't automated that should be? Are the sessions being stored? If yes, where, how, and why?
	Are users segmented into independent pods?
	What is the next scalability hurdle?

SECURITY	
	Are there approved and published security policies?
	How is sensitive information (e.g. passwords) stored, backed up, and transmitted?
	Does the system use HTTPS?
	Is there protection against well-known intrusion techniques?
	Are there approved and published security policies?
	Has penetration testing ever been performed?
	Who has root access?
	What are the upgrade and patching processes?
	What information is backed up and where?
	Has a disaster recovery test ever been performed?
	How could the site be more secure?

DEVELOPMENT PROCESSES	
	What languages and frameworks have been used?
	Who's on the development team?
	How is the team organized? How do they communicate and make decisions?
	Is the team cross-functional?
	Are team goals aligned to top level business goals?
	Do teams sit together?
	Is there revision control on all platforms/versions? If so, what is it?
	Is development test-driven?
	Does the system allow continuous integration? If so, which tools are used?
	How is deployment performed?
	How would you improve the development team?

MAINTAINABILITY	
	Is the source code readable and in a consistent style?
	Is the source code commented?
	Is there commenting for each code unit (e.g. object, method)
	Is the system running on current releases of underlying software?
	Are there any unexpected or obscure dependencies?
	How hard would it be to pick up and move to another platform (e.g. AWS)
	Are there any long-term viability issues with specific vendors?
	How could maintainability be improved?

LICENSING	
	Is all the code necessary to run the system properly licensed?
	How is the code licensed?
	Does anyone else have or claim legal rights to the code?
	What are the terms of the license(s)?
	Is there an IP strategy?

OPERATIONS	
	Are systems designed for monitoring?
	Are incidents logged with enough detail to troubleshoot potential problems?
	Are alerts sent in real time?
	Are user behaviors (e.g. logins, downloads, checkouts) used to create business metric monitors?
	Is remaining infrastructure headroom known?
	Are post-mortems conducted and fed back into the system?

OTHER	
	Are there any other interdependencies beyond your control?
	Any other pertinent information?

DISCLAIMER

Any articles, templates, or information provided by Smartsheet on the website are for reference only. While we strive to keep the information up to date and correct, we make no representations or warranties of any kind, express or implied, about the completeness, accuracy, reliability, suitability, or availability with respect to the website or the information, articles, templates, or related graphics contained on the website. Any reliance you place on such information is therefore strictly at your own risk.