Waste is anything that adds time without adding Value.

Form of Waste		Definition	Examples
	Defects	▪ Errors or deficiencies in work product.	▪ Code defect in production ▪ Security vulnerability ▪ Broken build ▪ Server misconfiguration
	Waiting	▪ Any delay that impedes value delivery	▪ Service Tickets ▪ Waiting on environments ▪ Slow build or test runs ▪ Batch processing
	Extra Features	▪ Feature bloat, over-design, duplicative work, or gold plating that is not required.	▪ Overly complex systems ▪ Unnecessary configuration settings ▪ Repeating processes
	Non-Utilized Talent	▪ A failure of the system to use human intelligence or allow smart decision making. Overuse of “Heroes”	▪ Rigid automation ▪ Single person bottleneck ▪ Overspecialization ▪ Broken standard process
	Re-Learning	▪ Failure to capture knowledge, especially around known issues or process, forcing repeat work to occur later	▪ Tribal knowledge ▪ Recurring issues not documented or repaired ▪ Documents created not used
	Handoffs	▪ Work hand off between individuals, organizational groups, or to contractors or third parties internal or external	▪ Any handoff between one team to another ▪ Handoffs between automated systems
	Task Switching	▪ A person in the value stream forced to stop work on one task to start a new, unrelated task	▪ Redefined Priorities ▪ Interruptions ▪ Engineers switch between multiple project teams
	Partially Done Work	▪ Work product that is partially done and must be finished or re-worked later in the value stream.	▪ Deployment script is wrong, Ops teams fixes it ▪ Feature not well defined, requiring redevelopment

Countermeasures are actions that reduce waste in the value stream

Form of Waste		Possible Organizational Countermeasures	Possible Automation Countermeasures
	Defects	▪ Increase feedback loops (builds, tests, scans) ▪ Training	▪ Code Scanning ▪ Automated Testing ▪ Automate System Verifications ▪ Access to prod-like environments
	Waiting	▪ Lower work in progress (WIP) ▪ Create slack in system to increase flow (lower resource utilization)	▪ Self service environments ▪ Minimize handoffs ▪ Automated approvals
	Extra Features	▪ Smaller features ▪ Time dedicated to refactoring and system enhancements ▪ Reduce handoffs	▪ Refactor system (Reduce technical debt) ▪ Eliminate duplicate processes
	Non-Utilized Talent	▪ Increase trust in technical teams ▪ Cross-train ▪ Decentralize or democratize	▪ Un-automate things that have high error rates ▪ “Andon cord” capability
	Re-Learning	▪ Update processes often ▪ Document tacit knowledge ▪ After action reviews	▪ Automate Runbook ▪ Wiki-like knowledge base ▪ ChatOps capability for sharing both system data and knowledge
	Handoffs	▪ Team re-organization by value stream ▪ Streamline process ▪ Cross-train staff ▪ Standardize work practices	▪ Single system across teams ▪ ChatOps and cross team communication tools ▪ Common source control for Dev and Infrastructure artifacts
	Task Switching	▪ Reduce defects, rework ▪ Minimize unplanned work ▪ Dedicated staff to teams ▪ Quiet Hours	▪ Use of Do Not Disturb settings ▪ Improvements to reduce defects, waiting, handoffs
	Partially Done Work	▪ Enforce standards at phases ▪ Checklists for completeness ▪ Lower Work in Process	▪ Automated acceptance testing ▪ Automate work advancement (e.g. reject builds where tests fail) ▪ Self service environments (fully done on demand)