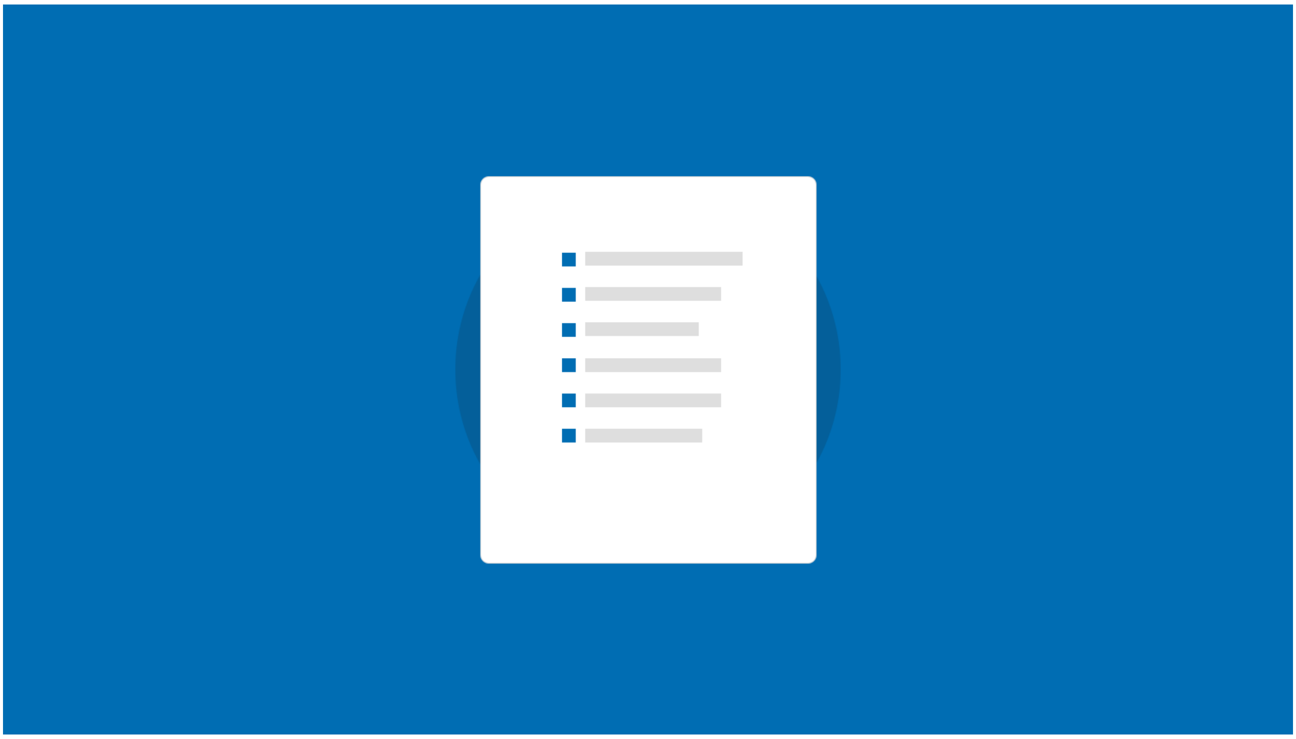
A Check List for Knowledge Transfer

Sooner or later, you have to deal with the task of project acceptance or transfer. To do that efficiently, I follow my own check list, so as not to lose sight of anything and make it in such a way that the customer or project owner would not notice the change of teams.



Team structure:

- [ ] Members of the team
- [ ] Hierarchy and responsibilities, reporting
- [ ] The customer's / related teams' / vendor's contacts

Business requirements, trying to get access to

- [ ] Description of business requirements
- [ ] User documentation
- [ ] Test cases

Source code:

- [ ] Repository URLs
- [ ] Creating accounts with the required rights
- [ ] Getting all configuration scripts and requirements for developers' workstations
- [ ] If possible, creating automated scripts of environment deployment or creating system images – to save precious time for the developers.

Technical documentation:

- [ ] System architecture
- [ ] Sub-system architecture
- [ ] Architecture in terms of technical primitives
- [ ] Architecture in terms of business tasks, use cases
- [ ] Team's technical debts
- [ ] All technical findings and proposals regarding the existing system
- [ ] It is possible to carry out an experiment: ask the new team to create a "pattern" of the system with the purpose of identifying infrastructure components that ensure the system's operability, and superimpose business requirements on that – judging by my experience, people catch on the project much quicker.

Delivery and environment system:

- [ ] CI/CD servers
- [ ] Test infrastructure
- [ ] Build versions

Information systems (very important):

- [ ] User requests handling system
- [ ] Bus tracking system
- [ ] Knowledge base system / information portal
- [ ] Monitoring system
- [ ] User actions analytics system
- [ ] It is crucial to identify contact persons for all issues regarding each system
- [ ] Also all procedures and ceremonials: flow processing of user requests; flow approvals of new technical documentation

Processes and a list of decision makers (DM):

- [ ] Procedures and a list of DMs connected with daily routines
- [ ] Procedures and a list of DMs connected with closing a sprint / iteration / work stage
- [ ] Procedures and a list of DMs connected with planning a new release / iteration
- [ ] Procedures connected with handling Change Requests, and a list of DMs
- [ ] Procedures and ceremonials connected with issuing a new release

Testing, quality assurance team:

- [ ] Necessary to get access to all test script databases
- [ ] Necessary to get the description of release issue procedures

User accounts – I bring them out, to keep in mind:

- [ ] Testing / staging / production user accounts for testing the product
- [ ] Accounts in the analytics / vendors / partners systems

Field of action:

- [ ] Work plan
- [ ] Roadmap
- [ ] Objectives definition in technical terms
- [ ] Objectives definition in terms of product

Third party services, vendors, partners:

- [ ] Access to all third party systems with admin rights – to be able to add your own users
- [ ] Contacts of all vendors and partners, their DMs, interaction schedule, their team structure, responsibilities, and a brief description of the interaction domain and the team's objectives

The "Objectives" Section is very important:

- [ ] What is required from the development, testing, and support teams?
- [ ] What are the objectives of the new team?