FORM 1A

Created on: 08 Feb 2022 8.55AM
Last updated on: 10 Feb 2022 2.18PM

ADMINISTRATION OF MUSLIM L
MUSLIM MARRIAGE AND I

Rule 3

فرموهنن انتوق اوفخارا دان فندفتران فرنكاحن

APPLICATION FOR SOLEMNISATION AND REGISTRATION

Tech lead checklist

Sensible defaults for effective delivery

/thoughtworks

Table of contents

Groom details

Full name نام فنوه

Alias نام لاءين

ID type جنيس ڤڠنلن

NRIC/ Passport number نمبر كارد ڤڠنلن / ڤسڤورت

Date of birth تاريخ لاهير

Marital status تارف ڤركهوينن †

Nationality كرعيتان / تارف فندودق

Race بڠسا

Religion

Tech lead checklist

Sensible defaults for effective delivery

Introduction

The purpose of this checklist is to help Tech Leads serve their team and lead a successful delivery. The checklist reflects Thoughtworks delivery methodology, based on a lean, agile and Extreme Programing (XP) delivery model that leverages hypothesis-driven design and development to continuously deliver thin slices of valuable working code. The delivery model for Thoughtworks engagements is typically broken down into three phases - Discovery, Inception and Delivery.

When we initiate a new engagement with our client, we seek to enter the engagement as early in the product lifecycle as possible, so that we can influence and shape the engagement as much as we can, to ensure we are set up for a successful delivery. The ideal entry point for an engagement is the Discovery phase, however, sometimes our clients are further along in their journey which means that we sometimes begin to partner with them in subsequent phases (Inception, Delivery).

This guide offers checklists for each phase to help set Thoughtworks expectations for Tech Leads across each phase of a typical engagement, and provide a starting point to build effective high performing teams.

Discovery

The Discovery phase is primarily a research phase where we seek to understand the customer's requirements, needs, and expectations. We develop hypotheses from current functionality, customers needs, market opportunities, and improvement opportunities that we prioritize for test-and-learn exploration activities. Many ideas and hypotheses are rejected as we identify the most valuable things on which to spend our time and effort. Discovery phases may optionally include client assessment(s).

through ambiguity, taking the lead on running key technical workshops

Declaration by Wali

Key Technical Workshops

Before scheduling key technical workshops it is important to kick off a Discovery by exploring the client's business problem first. The Tech Lead acts as a partner to the Thoughtworks Product Lead / Delivery Lead to jointly drive workshops that seek to understand the clients domain, business processes and identify high-level business outcomes (and quantitative measures of success) for the Thoughtworks engagement.

Key Technical Workshops:

- ☐ Architecture Overview
- ☐ Non-Functional / Cross-Functional Requirements Overview
- ☐ High-level Path to Production (Value Stream Map)
- ☐ Quality Overview
- ☐ Security Overview
- ☐ Delivery Infrastructure Overview
- ☐ Platform Capability Overview
- ☐ API Overview
- ☐ Data Overview
- ☐ Tooling Overview
- ☐ Operations & Production Support Overview

Note that for modernization engagements, there is usually a need to run Discovery workshops in the context of both current state and future state business processes and technical landscape.

On the completion of Discovery workshops (and optional assessment), the Tech Lead should drive the synthesis of raw technical workshop output (such as Mural boards)

into a consumable Readout for the purpose of client playback of Thoughtworks understanding of the problem space, sharing Thoughtworks recommendations and proposing next steps to progress the engagement to Inception and Delivery. The Readout might be a Discovery deliverable (part of our Statement of Work) and / or a lightweight final presentation as part of the Discovery wrap up. We recommend blocking time for internal synthesis after each workshop is completed so the Readout deck can be developed incrementally by the team while the context is still fresh, avoiding the unnecessary workload of pulling it all together in the final week.

Inception

The goal of an inception is to reach a shared understanding of the customer and business goals, tech vision and prioritized backlog to kick off a new workstream. The inception process allows the whole delivery team to be rapidly immersed in the context and domain. The shared knowledge which is established during the inception makes it easier to work with lower fidelity artifacts.

> **Inception is about taking the body of knowledge that was identified during the discovery phase and ensuring that everyone is focussed upon the activities needed to quick-start delivery. As a technical leader, your responsibility is to guide the team by taking a lean, evolutionary architecture approach that will support a "just-enough" solution. Furthermore, it is important to align this approach with the client stakeholders. This is critical because this conversation essentially establishes the scope of MVP (first thin slice of value delivered). Remember, inception is all about creating a shared vision.**

Key Technical Deliverables

> **Note that it is important to fully understand the functional scope derived from business needs and then build a technical strategy that solves the business problem. The Tech lead checklist is scoped to focus on the key technical deliverables that the team is accountable for in this context.**

- [ ] Technical Onboarding to remove developer friction during delivery
    - Pre-iteration Zero tasks & tickets - git repository, access, cloud account(s), tooling, local developer environment setup, and readme(s)
    - Glossary
- [ ] Iteration Zero planning
    - Technical spikes for the first few iterations to de-risk delivery
    - Foundational technical work (user story dependencies) to enable product delivery
    - Path-to-Production planning (Value Stream Map)
- [ ] Non-Funtional / Cross-Functional Requirements for the first thin-slice
- [ ] Socialize the evolving, high-level solution architecture (and supporting technical deliverables) as part of regular project showcases
- [ ] High-level Test Strategy
- [ ] High-level Security strategy with an account level Incident Response Plan
- [ ] High-level Platform Engineering Roadmap (if applicable)
- [ ] High-level Data Architecture (if applicable)

Delivery

The Delivery phase is the primary build phase where we deliver working code in thin slices of value. Typically, features are delivered in short iterations (normally 1-2 weeks) and released frequently. Ceremonies such as standups, showcases, and retrospectives are common activities driven by the Tech Lead to encourage fast feedback loops, demonstrate continuous value to the client, and lessons learned.

Team Rituals and Ways of Working

- [ ] Cultivate a team culture that encourages open communication, giving and receiving feedback timely and regularly and provides safety for all team members. Example activity could be regularly scheduled team wise speedback with individual follow ups encouraged.
- [ ] Encourage team bonding activities
- [ ] Pairing Matrix
- [ ] Introduce Tech Huddles during daily standup and Technical Roundtables (as needed)
- [ ] Collaborate with Project Managers, Product Managers, stakeholders and others during relevant team meetings (ie. planning, showcase preparation, etc.)
- [ ] Schedule Regular 1:1s with Thoughtworks team members and client stakeholders E.g. Product Partners, Delivery Partners, Project Managers, Engineering Managers, TLs, Architects, Engineers, CLT members
- [ ] Identify Thoughtworker performance partners for technical team members
- [ ] Align on team-level ways of working (core working hours, team rituals, social contract etc.)
- [ ] Manage team conflicts (technical and personal)
- [ ] Drive Software Engineering sensible default practices alignment
- [ ] Discourage managers from assigning user stories to developers. As an alternative, encourage team members to pick up user stories. This promotes

accountability because they won't feel like they have to be told what to do all the time.

- [ ] Drive Technical Debt management
- [ ] Act as a scout to identify and manage cross team (technical) dependencies

Production Readiness

As your team is working on delivering features for the product, it's important, as a Tech Lead, to be aware and advocate for what it takes to deliver effectively and be production ready. Concepts such as continuous integration, code quality validations and security scanning are key to making sure the team is production ready.

- [ ] Path to production story breakdown and incremental delivery plan (technical roadmap)
- [ ] Operations / Production Support schedules, dashboards & alerts
- [ ] Create Delivery Pipeline that builds & runs automated test suite such as unit tests, component tests, contract tests
- [ ] Run security scan in delivery pipeline (SAST & SCA)
- [ ] Deploy Infrastructure (IaC) and application(s) to all environments automatically using pipeline
- [ ] Continuous integration and continuous delivery, deploying to production early and often
- [ ] Baseline and continuously measure 4 Key Metrics

Technical Vision

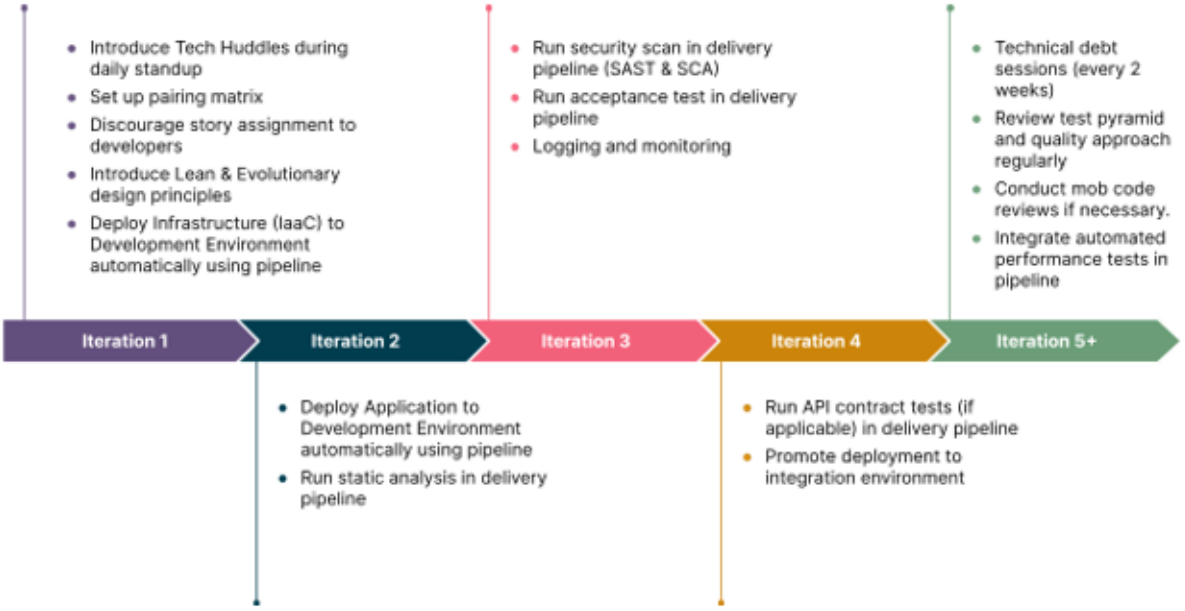
- [ ] API Strategy

- [ ] Introduce Lean & Evolutionary Architecture design principles

Quality Engineering

- [ ] Test-driven Development
- [ ] Run static analysis in delivery pipeline
- [ ] Test Strategy, including review of existing test pyramid and quality approach
- [ ] Discuss and align on code review practices
- [ ] Pair programming
- [ ] Consider Mob Code reviews

Example Incremental Delivery Technical Roadmap



As a Tech Lead, one of your core objectives is to guide the team to deliver high quality working code to production early and often. The above diagram visualizes an example of an incremental technical delivery roadmap for the path to production by incorporating cross-functional requirements (CFRs) in small batches. It is always

advisable to sprinkle in CFRs such as security, observability and performance testing throughout all delivery iterations as opposed to waiting until the very end (or having dedicated "hardening" iterations). If there is a lot of foundational technical effort required to kick off product delivery, it may be worthwhile to frontload some of the path to production technical tasks into Iteration 0. Once feature work starts in iteration 1, it is recommended practice to bake technical requirements into the business feature stories so that the architectural-ilities that are required for production readiness are delivered at a steady pace.

Helpful resources

- [ ] The Thoughtworks Guide - A collection of our practices and processes
- [ ] Sensible Default Practices - Commonly agreed approaches for the ways we work serve to improve operational efficiency, reduce friction and increase cohesion for Thoughtworks teams located in different geographic regions when working with clients.
- [ ] Team Collaboration Anywhere - A collection of resources for effectively leading hybrid teams (mix of remote and in-person teams)

Where to get additional help

As a starting point, talk to your Account Technical Principal to discuss any gaps, concerns or necessary support needed for your client engagement.

To share feedback on the Tech lead checklist, please reach out to:

- North America Learning & Development: na-L&D@thoughtworks.com
- North America Head of Technology: na-hot@thoughtworks.com

Acknowledgements

Charith Tangirala created the Tech lead checklist, in collaboration with the NA HoT team and NA L&D team. Our thanks to Brandon Cook, Erica Kastner, Kieran Murphy, Lina Qu, Marco Valtas, Premanand Chandrasekaran, Punit Lad, and Vanessa Towers for their feedback and contributions.

Title	Published	Notes
Tech lead checklist - sensible defaults	Jan 26th, 2023	First Version

/thoughtworks