TTS Project: Thai Language – End-to-End Program Management

1. Project Overview

The TTS (Text-to-Speech) project for the Thai language aimed to deliver high-quality voice assets for integration into consumer devices. The end-to-end scope involved planning, data acquisition, recordings, asset creation, modeling, validation, testing, and deployment. A custom deep learning-based pronunciation engine, 'Hydra', was also used to handle Thai-specific phonetic complexity.

2. Project Management Framework

2.1 Initiation

- Defined project goals and scope.
- Identified stakeholders across engineering, linguistics, QA, and crowdsourcing teams.
- Created a high-level project charter and got stakeholder buy-in.

2.2 Planning

- Created a detailed roadmap (see below).
- Defined scope, schedule, resources, and risk management strategies.
- Outlined communication plans and milestones.
- Communication with stakeholders and engineering teams was crucial early in planning. Since the Thai language TTS was built from scratch, the initial release goals were not feasible within current constraints. The TPM led negotiation efforts, proposing a phased delivery: server-based assets would be delivered at release, and device-ready assets would follow 4 sprints later.
- This proposal was accepted by both stakeholders and the engineering team, and incorporated into the official planning, demonstrating agile adaptability beyond traditional project management structure.

Project Roadmap

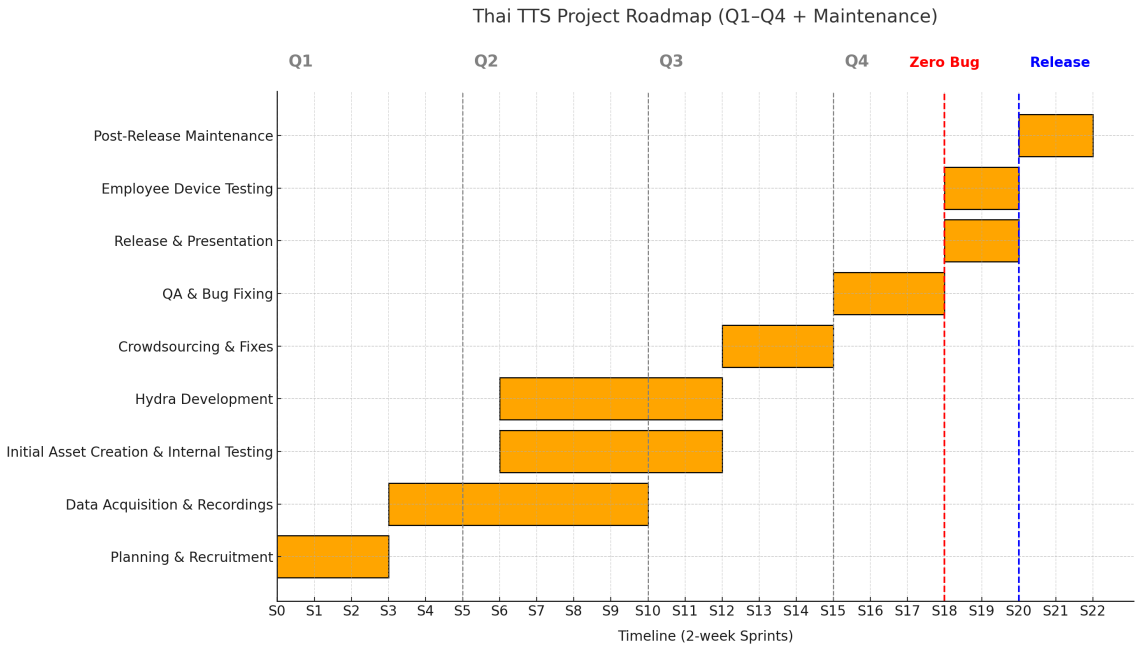

Figure: Thai TTS Project Roadmap – Final Structure (Q1–Q4 + Maintenance)

2.3 Execution

- Hired Language Engineer, Linguist to assist in the studio, and Voice Talents.
- Recordings were conducted by the Studio. The TTS team was responsible for creating scripts, following up on progress, and performing post-recording QC.
- Used pre-built Hydra (pronunciation model) and FastSpeech2 (TTS model) systems developed by the Researcher team.
- Focused on configuring and adjusting the training pipeline to generate quality TTS assets.
- Iterated with different settings including learning rates, number of epochs, pre-training and fine-tuning steps.
- Selected candidate assets for external validation via crowdsourcing followed by focused fixes.

2.4 Monitoring & Controlling

- Conducted a two-sprint phase for crowdsourcing evaluation and adjustments, using structured A/B testing to gather feedback.
- Applied fixes and re-tested assets.
- Ran a compatibility test between updated assets and the live system after QA.
- Since individual assets could be changed without replacing the entire model, this compatibility test checked integration reliability.
- Key system parameters such as latency were evaluated to ensure proper performance.
- Sent to QA with defined thresholds and scoring expectations.
- QA returned issues to be addressed and a ‘zero-bug’ policy was enforced before transitioning to the release phase.
- Bugs from employee testing were tracked and resolved within the release window.

2.5 Closing

- Conducted release presentation covering scores, model parameters, and production-readiness.
  * Average MOS score: 4.5/5 from internal evaluations.
  * Pronunciation accuracy (Hydra engine): 97.2%.
  * Bug resolution rate pre-release: 100% compliance with 'zero-bug' policy.
  * Time to QA clearance: 3 sprints.
  * Employee-reported issue rate during testing phase: <2%.
- Formed post-release maintenance team for error prevention and brand protection.
- Archived documentation and completed project post-mortem.

3. Technical Highlights

- Hydra (pronunciation model) and FastSpeech2 (speech synthesis model) were developed by the Researcher team.
- The Language Engineer focused on tuning the pipeline with optimal configurations.
- Parameters such as learning rate, iterations, pre-training stages, fine-tuning depth, and epoch count were adjusted iteratively.
- Pronunciation quality was evaluated using a Pronunciation Validation (PV) test.
- Integration with phonetic transcription APIs ensured automation and scalability.
- Used AWS S3 for storage, Bitbucket for front-end versioning (FE: pronunciation), GitHub for backend management (BE: data), and Docker to deploy and run microservices.

4. Key Deliverables

- Voice assets for TTS integration.
- Hydra pronunciation system.
- QA reports and test results.
- Post-release error handling plan.
- Full project documentation and roadmap.

5. Budget Considerations

The project required careful planning and control of budget allocation, especially for crowdsourcing phases. Budget was allocated for external voice talent recording (via Studio), human validation (crowdsourcing), and cloud services (AWS S3, Docker deployments). Coordination with finance and procurement teams ensured proper allocation, tracking, and approval processes for each spending phase.