Task 1: Agile Development Plan & IT Specialist Responsibilities

Last updated: 2025-10-23T07:31:43+03:00

Prepared for: Elektrum Eesti OÜ - IT Projektijuht Position

Date: October 23, 2025

Author: Mihkel Putrinš

Executive Summary

The EE HH Recommendation functionality represents a strategic customer acquisition initiative that leverages existing customer satisfaction to drive new contract growth. The system enables customers to share unique promo codes with prospects, creating a win-win scenario where both parties receive financial incentives when new contracts are activated.

This document outlines a structured Agile implementation approach spanning 4 sprints (8 weeks) with clear milestones, deliverables, and IT specialist responsibilities. The solution integrates with CC&B billing system, self-service portal (Elektrum.ee), Sales Portal, and mobile platforms to create a seamless user experience.

Key Success Metrics:

- Customer acquisition cost reduction through recommendation channel
- Promo code adoption rate among existing customers
- New customer conversion rate via promo codes
- System reliability (99.5% service availability target)

---

1. Agile Development Plan

Sprint 0: Planning & Foundation (Week 1)

Objectives:

- Finalize requirements and acceptance criteria
- Establish development environment and tools
- Create initial product backlog
- Define team structure and communication protocols

Key Activities:

1. **Requirements Workshop** (Days 1-2)
   - Deep-dive into specification document with business stakeholders
   - Clarify promo code generation rules and lifecycle
   - Validate discount calculation logic and group definitions
   - Map integration touchpoints (CC&B, portals, mobile)
   - Document business rules and edge cases
2. **Technical Architecture Design** (Days 2-3)
   - Database schema design (promo code storage, tracking tables)
   - API design for code generation, validation, discount application
   - Integration patterns with CC&B and portal systems
   - Security considerations (code uniqueness, validation, fraud prevention)
   - Performance planning (code lookup optimization, caching strategy)

3. **Product Backlog Creation** (Days 3-4)
    - Break down features into user stories with acceptance criteria
    - Estimate story points using Planning Poker
    - Prioritize based on business value and technical dependencies
    - Identify technical spikes for research items
4. **Team Setup** (Day 5)
    - Development environment configuration
    - CI/CD pipeline setup
    - Code repository and branching strategy
    - Testing framework initialization
    - Documentation structure

Deliverables:

- Requirements specification document (validated with stakeholders)
- Technical architecture diagram
- Product backlog (prioritized, estimated)
- Definition of Done criteria
- Sprint 1 commitment and plan

IT Specialist Responsibilities in Sprint 0:

- Lead requirements gathering sessions with business analysts
- Document functional and non-functional requirements
- Create user stories with clear acceptance criteria
- Coordinate with CC&B development team on integration approach
- Set up test environment and data

Sprint 1: Promo Code Generation & Storage (Weeks 2-3)

Sprint Goal: Enable automatic promo code generation for existing and new customers with persistence in CC&B.

User Stories:

1. **US-001**: As an existing electricity customer, I want a unique promo code automatically generated when my contract activates, so I can share it with friends.
    - **Acceptance Criteria**:
        - Code generated on contract activation event
        - Code is unique across all accounts
        - Code stored in CC&B Account → Services tab
        - Code format: 8-character alphanumeric (e.g., "A123B456")
        - Code status: ACTIVE upon generation
2. **US-002**: As a system administrator, I want promo codes stored in CC&B with status tracking, so I can monitor code lifecycle.
    - **Acceptance Criteria**:
        - Database table created with fields: code, account_id, status, start_date, end_date, created_at
        - Indexes on code (unique) and account_id for fast lookup
        - Status values: ACTIVE, INACTIVE, EXPIRED
        - Audit trail for code status changes
3. **US-003**: As an IT specialist, I want manual promo code generation capability in CC&B, so I can handle edge cases (failed generation, code issues).

- **Acceptance Criteria**:
  - Manual generation button in CC&B Account → Services tab
  - Validation prevents duplicate codes
  - Log entry created for manual generation (audit trail)
  - Generated code follows same format and rules as automatic

Key Activities:

- Database migration scripts for promo code tables
- Promo code generation service implementation (uniqueness algorithm)
- CC&B integration (event listener for contract activation)
- Manual generation UI in CC&B Services tab
- Unit tests (code generation, uniqueness validation)
- Integration tests (CC&B event handling)

Sprint Ceremonies:

- **Sprint Planning** (Day 1): Story selection, estimation, commitment
- **Daily Stand-ups** (15 min): Progress updates, blockers, collaboration needs
- **Sprint Review** (Last day, PM): Demo to stakeholders, gather feedback
- **Sprint Retrospective** (Last day, afternoon): Process improvements, lessons learned

IT Specialist Responsibilities in Sprint 1:

- Coordinate with CC&B team on database schema changes
- Review and approve code generation algorithm (uniqueness, format)
- Test integration with CC&B contract activation events
- Document API endpoints and data structures
- Create test data for QA team
- Triage and prioritize bugs found during sprint

Sprint 2: Self-Service Portal & Code Display (Weeks 4-5)

Sprint Goal: Enable existing customers to view and share promo codes via self-service portal and mobile apps.

User Stories:

1. **US-004**: As an existing customer, I want to see my promo code on Elektrum.ee self-service portal, so I know what code to share.
   - **Acceptance Criteria**:
     - Promo code displayed in customer account dashboard
     - Clear instructions on how to share code
     - "Copy to clipboard" functionality
     - Code shown with status indicator (Active/Inactive)
2. **US-005**: As an existing customer, I want a pre-filled sharing message with my code, so I can easily share via email, WhatsApp, or SMS.
   - **Acceptance Criteria**:
     - "Share" button generates pre-filled text: "Join Elektrum using my code [CODE] and we both get [X] EUR discount!"
     - Share options: email, WhatsApp, SMS, copy to clipboard
     - Share tracking (optional): log when share button is clicked
     - Works on both web (desktop/mobile) and mobile apps (Android/iOS)

3. **US-006**: As an existing customer, I want to see how many times my code was used, so I can track my successful recommendations.
   - **Acceptance Criteria**:
     - Display: "Your code has been used X times"
     - Show recent successful recommendations (last 3-5, with dates)
     - Display earned discount total from recommendations

Key Activities:

- Self-service portal UI development (React/Vue components)
- API integration with promo code service
- Mobile app updates (Android/iOS native or React Native)
- Share functionality implementation (native share APIs)
- Responsive design for mobile web
- E2E tests for portal flows

IT Specialist Responsibilities in Sprint 2:

- Review UI/UX designs with portal team
- Test API integration between portal and promo code service
- Coordinate mobile app updates with mobile development team
- Validate sharing functionality across platforms (web, Android, iOS)
- Test responsive design on various devices
- Document user-facing features for support team

Sprint 3: Code Usage & Discount Application (Weeks 6-7)

Sprint Goal: Enable new customers to use promo codes during signup and automatically apply discounts to both parties.

User Stories:

1. **US-007**: As a new customer, I want to enter a promo code when signing up via self-service, so I can get a discount.
   - **Acceptance Criteria**:
     - Promo code input field in contract signup flow
     - Real-time validation (code exists, is active)
     - Display discount amount if code is valid
     - Show error message if code is invalid/inactive
     - Promo code optional (signup works without it)
2. **US-008**: As a new customer, I want to see the discount amount before completing signup, so I know what benefit I'll receive.
   - **Acceptance Criteria**:
     - Display: "With code [CODE], you'll get [X] EUR discount"
     - Discount amount based on effective discount group at signup moment
     - Clear explanation of when discount is applied (after contract activation)
3. **US-009**: As the system, I want to automatically apply discounts to both customers when new contract activates, so the recommendation reward is processed correctly.
   - **Acceptance Criteria**:
     - Trigger: new customer contract activation event
     - Validate promo code is still active at activation
     - Apply discount to new customer account (discount group: new customer self-service)

    - Apply discount to existing customer account (discount group: existing customer)
    - Use "Continuous bill discount" type (as specified in LEO-25125)
    - Log both discount applications for audit
4. **US-010**: As a Sales Portal employee, I want to manually enter promo codes for customers signing up via phone/office, so offline channels can also benefit.
    - **Acceptance Criteria**:
        - Promo code field in Sales Portal sales event form
        - Validation with warning (not blocking) if code inactive
        - Ability to override discount group if needed (e.g., customer started online but finished offline)
        - Discount amount displayed after code entry

Key Activities:

- Self-service portal signup flow updates (add promo code field)
- Promo code validation API development
- Discount application service (integration with billing system)
- Sales Portal updates (promo code field, validation)
- Business logic implementation (discount groups, effective dates)
- Automated tests for discount calculation and application

IT Specialist Responsibilities in Sprint 3:

- Validate discount calculation logic with business stakeholders
- Test promo code validation across all entry points (self-service, Sales Portal)
- Verify discount application matches specification (LEO-25125 reference)
- Coordinate with billing team on "Continuous bill discount" implementation
- Test edge cases (inactive codes, expired discounts, duplicate usage attempts)
- Document promo code usage flows for operations team

Sprint 4: Edge Cases, Reporting & Launch (Week 8)

Sprint Goal: Handle edge cases, implement tracking/reporting, finalize documentation, and prepare for production launch.

User Stories:

1. **US-011**: As a system administrator, I want to manually inactivate promo codes (e.g., fraud prevention), so I can block misuse.
    - **Acceptance Criteria**:
        - "Inactivate" button in CC&B for each promo code
        - Reason field required (e.g., "Suspected fraud", "Customer request")
        - Inactivated codes cannot be used for new signups
        - Existing discounts from this code remain valid
        - Audit log entry created
2. **US-012**: As a business analyst, I want a report of promo code usage (sent/used, conversion rates), so I can measure program effectiveness.
    - **Acceptance Criteria**:
        - Report shows: total codes generated, codes shared, codes used, conversion rate
        - Filter by date range, customer segment, discount group
        - Export to CSV/Excel
        - Accessible via CC&B reporting module

3. **US-013**: As a customer, I want unused discount transferred if I close my account, so I don't lose my earned rewards.
    - **Acceptance Criteria**:
        - If customer has multiple accounts, show option to transfer discount
        - Transfer to account selection (dropdown of customer's other accounts)
        - Confirmation dialog before transfer
        - Discount amount and type preserved
        - Original account discount marked as transferred (audit trail)

Key Activities:

- Edge case handling (contract termination, account transfers, fraud scenarios)
- Reporting dashboard development (analytics, metrics, exports)
- End-to-end testing (complete user journeys)
- User acceptance testing (UAT) with business stakeholders
- Performance testing (load test code generation, validation, discount application)
- Security audit (code guessing, brute force prevention)
- Documentation finalization (user guides, operations runbooks, API docs)
- Deployment planning (rollout strategy, rollback plan)

IT Specialist Responsibilities in Sprint 4:

- Coordinate UAT sessions with business users
- Document edge cases and their handling
- Create operations runbook for support team
- Prepare rollback plan if critical issues arise
- Conduct performance review and optimization
- Final sign-off on deployment readiness

Release & Post-Launch (Week 9+)

Release Activities:

1. **Deployment** (Weekend or low-traffic period)
    - Database migrations in production
    - Application deployment (blue-green or canary)
    - Smoke tests in production
    - Communication to customers (new feature announcement)
2. **Monitoring** (First 2 weeks post-launch)
    - System health metrics (uptime, response times, error rates)
    - Business metrics (code generation rate, usage rate, discount application success)
    - User feedback collection (support tickets, feedback forms)
    - Daily check-ins with operations team
3. **Retrospective** (End of Week 9)
    - What went well in the project?
    - What could be improved?
    - Lessons learned for future projects
    - Process improvements to implement

IT Specialist Responsibilities Post-Launch:

- Monitor system performance and error logs
- Triage production issues and coordinate fixes
- Collect and prioritize feedback for future iterations

- Document lessons learned
- Support operations team with questions/issues
- Plan next iteration (e.g., expanding to business customers, additional features)

---

2. IT Specialist Responsibilities

The IT Specialist serves as a bridge between business stakeholders and technical teams, ensuring requirements are clearly understood, solutions are correctly implemented, and quality standards are met throughout the project lifecycle.

2.1 Requirements Phase

Primary Responsibilities:

- **Business Requirements Gathering**
  - Conduct workshops with business stakeholders to understand promo code system goals
  - Document functional requirements (code generation, sharing, usage, discount application)
  - Document non-functional requirements (performance, security, scalability, availability)
  - Clarify edge cases (contract termination, fraud scenarios, system failures)
  - Validate requirements with stakeholders before design phase
- **User Story Creation**
  - Break down features into user stories with clear acceptance criteria
  - Ensure stories follow INVEST principles (Independent, Negotiable, Valuable, Estimable, Small, Testable)
  - Prioritize stories based on business value and technical dependencies
  - Maintain product backlog with stakeholder input
- **Integration Analysis**
  - Map integration points with existing systems (CC&B, portals, mobile apps)
  - Document data flows between systems
  - Identify API requirements and data format specifications
  - Assess integration complexity and risks

Deliverables:

- Business requirements document
- Product backlog with prioritized user stories
- Integration architecture diagram
- Risk assessment and mitigation plan

2.2 Design & Planning Phase

Primary Responsibilities:

- **Technical Coordination**
  - Facilitate design discussions between development teams (CC&B, portal, mobile)
  - Review technical architecture and provide feedback
  - Ensure design aligns with business requirements
  - Validate security and performance considerations
- **Sprint Planning Support**

  - Participate in sprint planning sessions
  - Clarify story requirements and acceptance criteria
  - Help estimate story complexity
  - Identify dependencies between stories and sprints
- **Test Planning**
  - Define test strategy (unit, integration, E2E, UAT, performance)
  - Create test scenarios based on user stories and edge cases
  - Plan UAT approach with business stakeholders
  - Coordinate with QA team on test environment setup

Deliverables:

- Technical design review notes
- Sprint plans with story commitments
- Test strategy document
- UAT plan with business stakeholder involvement

2.3 Development Phase

Primary Responsibilities:

- **Daily Coordination**
  - Attend daily stand-ups to track progress and blockers
  - Clarify requirements when questions arise
  - Facilitate communication between business and technical teams
  - Escalate blockers that require business decisions
- **Code Review & Quality Assurance**
  - Review code for alignment with requirements (not deep technical review)
  - Validate implemented features against acceptance criteria
  - Test features in development environment
  - Provide feedback to developers on business logic correctness
- **Blocker Resolution**
  - Identify and resolve requirement ambiguities
  - Coordinate with business stakeholders for clarifications
  - Facilitate technical discussions when integration issues arise
  - Track and communicate risks to project manager
- **Documentation**
  - Maintain up-to-date requirements documentation
  - Document requirement changes and their impact
  - Update user stories based on implementation learnings
  - Create draft user guides and FAQ content

Deliverables:

- Daily status updates to stakeholders
- Requirement clarification notes
- Updated user stories and acceptance criteria
- Draft user documentation

2.4 Testing Phase

Primary Responsibilities:

- **Test Case Design**
  - Create detailed test cases for integration testing
  - Define test data scenarios (happy path, edge cases, error conditions)
  - Document expected results for each test case
  - Ensure test coverage for all acceptance criteria
- **QA Coordination**
  - Work with QA team to execute test cases
  - Validate test results against acceptance criteria
  - Triage bugs (severity, priority, root cause analysis)
  - Decide which bugs block release vs. can be deferred
- **User Acceptance Testing (UAT)**
  - Coordinate UAT sessions with business stakeholders
  - Prepare UAT environment and test data
  - Guide stakeholders through test scenarios
  - Collect and document feedback
  - Validate that implemented solution meets business needs

Deliverables:

- Test cases and scenarios
- UAT plan and schedule
- Bug triage reports
- UAT sign-off document

2.5 Deployment Phase

Primary Responsibilities:

- **Release Coordination**
  - Participate in deployment planning (timing, rollout strategy)
  - Validate deployment checklist (database migrations, configuration, smoke tests)
  - Coordinate communication to customers and support team
  - Monitor deployment process and be available for issues
- **Smoke Testing**
  - Execute critical path tests in production immediately after deployment
  - Validate promo code generation, validation, and discount application
  - Verify integration with CC&B and portals is working
  - Confirm no major regressions in existing functionality
- **Rollback Planning**
  - Ensure rollback plan is documented and tested
  - Define rollback criteria (what issues trigger rollback?)
  - Coordinate with operations team on rollback procedure
  - Communicate rollback decision to stakeholders if needed

Deliverables:

- Deployment checklist
- Smoke test results
- Rollback plan document
- Deployment communication to stakeholders

2.6 Post-Deployment & Maintenance Phase

Primary Responsibilities:

- **Production Monitoring**
  - Monitor system health metrics (error rates, response times, uptime)
  - Track business metrics (code generation rate, usage, discount application)
  - Review error logs and identify patterns
  - Escalate critical issues immediately
- **Incident Management**
  - Triage production issues reported by users or monitoring
  - Coordinate with development team on bug fixes
  - Communicate status to stakeholders
  - Document incident and resolution for knowledge base
- **Continuous Improvement**
  - Collect user feedback from support tickets and feedback forms
  - Analyze metrics to identify improvement opportunities
  - Prioritize enhancements and bug fixes for next iteration
  - Facilitate retrospectives and implement process improvements
- **Knowledge Transfer**
  - Create and maintain operations runbook for support team
  - Train support staff on promo code system functionality
  - Document known issues and workarounds
  - Update user guides based on real-world usage patterns

Deliverables:

- Production monitoring reports
- Incident response documentation
- Enhancement backlog for next iteration
- Operations runbook and training materials

---

3. Key Success Factors

3.1 Communication & Collaboration

- **Stakeholder Engagement**: Regular touchpoints with business stakeholders throughout project
- **Team Coordination**: Daily communication with development, QA, and operations teams
- **Transparency**: Proactive communication of risks, blockers, and status changes
- **Feedback Loops**: Continuous validation that implementation matches business expectations

3.2 Agile Discipline

- **Iterative Delivery**: Working software delivered every 2 weeks (sprint cadence)
- **Flexibility**: Adapt to changing requirements while maintaining focus on sprint goals
- **Retrospectives**: Continuous process improvement based on team learnings
- **Definition of Done**: Clear quality gates ensure consistent delivery standards

3.3 Quality Assurance

- **Test-First Mindset**: Acceptance criteria defined before development begins

- **Multi-Layer Testing**: Unit, integration, E2E, UAT, performance testing
- **Edge Case Coverage**: Explicit handling of error scenarios and edge cases
- **Production Readiness**: Thorough validation before go-live

3.4 Risk Management

- **Early Identification**: Proactive risk assessment in Sprint 0 and throughout project
- **Mitigation Planning**: Clear strategies for addressing identified risks
- **Contingency Plans**: Rollback and workaround procedures documented
- **Monitoring**: Post-launch monitoring to catch issues early

---

4. Conclusion

The Agile development plan outlined above provides a structured, iterative approach to delivering the EE HH Recommendation functionality. By breaking the project into 4 focused sprints with clear goals and deliverables, we ensure:

- **Business Value**: Each sprint delivers working features that can be validated with stakeholders
- **Risk Mitigation**: Early integration testing and incremental delivery reduce big-bang release risks
- **Quality**: Multi-phase testing (unit, integration, UAT) ensures robust implementation
- **Flexibility**: Agile ceremonies (sprint reviews, retrospectives) allow for course correction

The IT Specialist role is critical throughout this journey, serving as the bridge between business needs and technical implementation, ensuring requirements are clearly understood, solutions are correctly implemented, and quality standards are maintained from concept through production.

Next Steps:

1. Stakeholder approval of this plan
2. Team capacity planning and assignment
3. Sprint 0 kickoff and backlog refinement
4. Development environment setup
5. Sprint 1 launch (Week 2)

Document Version: 1.0

Date: October 23, 2025

Author: Mihkel Putrinš

Status: Ready for Review