

Reference Guide for Software Development Project Management

Emmanuel BRUNO
Université de Toulon, LIS UMR CNRS 7020
emmanuel.bruno@univ-tln.fr

2026-10-14

Table des matières

1 Introduction to Development Methodologies	1
2 Course Structure	1
3 Core Agile Methodologies	1
3.1 Scrum Framework	1
3.2 Kanban Method	2
3.3 Extreme Programming (XP)	2
3.4 Ceremony Comparison	2
3.5 Implementation Guidelines	2
3.6 Team Structure and Roles	2
3.7 Role-Specific Responsibilities	2
3.8 Planning Approaches by Methodology	2
3.9 Process Workflow	2
3.10 Release Planning	3
3.11 Agile Best Practices	3
3.12 Acceptance Criteria and Definition of Done	3
3.13 Ceremony Details	3
3.14 Scaling Agile Frameworks	3
3.15 Risk Management in Agile	3
3.16 Quick Start Guide	4

1 Introduction to Development Methodologies

This document presents a reference framework for software development project management. It covers the main methodologies,

analysis techniques, and design approaches recommended for modern projects.

2 Course Structure

1. Foundations

- Agile Frameworks
- Ceremonies and Meetings
- Team Roles and Responsibilities

2. Implementation

- Project Planning
- Requirements Management
- Tools and Technologies

3. Advanced Topics

- Metrics and KPIs
- Scaling Agile
- Documentation Practices
- Risk Management

3 Core Agile Methodologies

3.1 Scrum Framework

- Time-boxed iterations (sprints)
- Fixed roles (Scrum Master, Product Owner, Dev Team)
- Structured ceremonies and artifacts
- Best for: Projects with defined scope and stable teams
- Detailed Ceremonies Guide
- Team Roles Guide

3.2 Kanban Method

- Continuous flow delivery
- Visual workflow management
- WIP limits and pull system
- Best for: Support, maintenance, and variable workload
- Ceremonies Guide
- Team Roles Guide

3.3 Extreme Programming (XP)

- Technical excellence focus
- Pair programming and TDD
- Continuous integration/deployment
- Best for: Complex projects requiring high quality
- XP Ceremonies Guide

3.4 Ceremony Comparison

Scrum (Details): - Sprint Planning (2-4 hours)
- Daily Scrum (15 min) - Sprint Review (1-2 hours) - Sprint Retrospective (1.5 hours)

Kanban (Details): - Daily Standup (15 min) - Flow Reviews (1 hour) - Operations Review (2 hours) - Replenishment Meetings (as needed)

XP (Details): - Planning Game (Weekly) - Weekly Cycle Review - Quarterly Cycle Planning - Stand-up Meetings (Daily)

3.5 Implementation Guidelines

Team Structure: - Scrum: 5-9 dedicated members - Kanban: Flexible team size - XP: Small co-located teams

Planning Approach: - Scrum: Fixed sprints (2 weeks) - Kanban: Continuous flow - XP: Weekly/Quarterly cycles

3.6 Team Structure and Roles

Scrum Teams (Details): - Product Owner: Vision and backlog management - Scrum Master: Process facilitation - Development Team: Cross-functional, 5-9 members - Key ceremonies: Sprint Planning, Daily Scrum, Reviews

Kanban Teams (Details): - Service Delivery Manager: Flow optimization - Flow Manager: WIP limits and board management - Development Team: Self-organizing, flexible size - Focus on continuous flow and delivery

Common Development Roles: - Software Developers - Quality Assurance Engineers - UI/UX Designers - Technical Leads

3.7 Role-Specific Responsibilities

Technical Skills Requirements: - Agile practices proficiency - Technical expertise levels - Communication abilities - Planning capabilities - Testing competencies

RACI Matrices: - Available in methodology-specific guides - Clear responsibility definitions - Cross-functional accountability - Stakeholder involvement patterns

3.8 Planning Approaches by Methodology

Each agile methodology has its specific planning practices. For detailed planning guidelines, see our Agile Planning Guide.

Scrum Planning: - Fixed-length sprints (typically 2 weeks) - Sprint planning meetings (2-hour timeboxed) - Story points estimation - Velocity-based capacity planning

Kanban Planning: - Continuous flow planning - WIP limits by column - No fixed iterations - Service class prioritization

XP Planning: - Weekly iterations - Release planning (quarterly) - Close customer collaboration - Technical spike planning

3.9 Process Workflow

Standard Development States: 1. Backlog Management - Grooming and refinement activities - Quality gates and acceptance criteria 2. Sprint Planning & Estimation - Story point estimation - Capacity planning 3. Active De-

velopment - Daily progress tracking - Sprint metrics monitoring 4. Review & Testing 5. Completion & Deployment

3.10 Release Planning

Recommended Structure: - MVP (Minimum Viable Product): Core features only - Incremental releases: 4-6 weeks per release - Priority reassessment after each user feedback

3.11 Agile Best Practices

Communication: - Daily standup: 15 minutes max, focused on progress - Regular retrospectives: Bi-weekly recommended - Living documentation: Continuous updates, accessible to all

Recommended Tools: - Project management: Jira, Trello, Azure Boards - Documentation: Confluence, GitBook - Collaboration: Slack, Microsoft Teams - Version control: Git with GitHub/GitLab For detailed tool comparisons and selection criteria, see our Tool Selection Guide.

3.12 Acceptance Criteria and Definition of Done

Quality assurance is crucial in agile development. See our comprehensive Definition of Done guide for detailed criteria.

Work Item Levels (Details): - Epic Level: System-wide integration and business metrics - Feature Level: Integration testing and stakeholder approval - User Story Level: Code quality, testing, documentation - Sprint Level: Potentially releasable increment - Release Level: Full regression and compliance checks

Methodology-Specific Requirements: - Scrum (Details): - Sprint and Release level validations - Story point validation against velocity - Ceremony completion requirements - Kanban (Details): - WIP limits and flow metrics - Lead/cycle time measurements - Continuous

quality gates - XP (Details): - Pair programming verification - TDD compliance - Zero technical debt policy

Quality Gates (Details): - Code Quality: Static analysis, security scans - Build Quality: Test coverage, dependency checks - Deployment Quality: Performance metrics, security - SonarQube Metrics: Reliability, maintainability

3.13 Ceremony Details

Our Scrum Ceremonies Guide provides comprehensive information about:

Core Ceremonies: - Sprint Planning (2-4 hours) - Daily Scrum (15 minutes) - Sprint Review (1-2 hours) - Sprint Retrospective (1-1.5 hours)

Remote Guidelines: - Mandatory video participation - Virtual board utilization - Strict timeboxing - Documentation standards

3.14 Scaling Agile Frameworks

For detailed scaling guidelines, see our Scaling Guide.

Common Frameworks: - SAFe: Enterprise-scale (100+ members) - LeSS: Basic (2-8 teams) and Huge (8+ teams) - Nexus: 3-9 Scrum teams - Scrum@Scale: Modular scaling

Key Implementation Areas: - Team of Teams structure - Communities of Practice - Cross-team coordination - Shared services model

3.15 Risk Management in Agile

For comprehensive risk management strategies, refer to our Risk Management Guide.

Risk Categories: - Technical: Architecture, debt, security - Process: Requirements, scope, estimation - People: Skills, resources, conflicts - Business: Market, budget, compliance

Mitigation Approaches: - Sprint-level risk reviews - Risk-adjusted backlog prioritization - Continuous assessment - Early warning indicators

3.16 Quick Start Guide

1. Methodology Selection

- Evaluate project characteristics
- Consider team size and location
- Assess technical requirements
- Match delivery expectations

2. Tools Setup

- Configure work environment
- Define Git workflows
- Establish code standards

3. First Steps

- Define MVP
- Create initial backlog
- Plan first sprint/iteration

Success Indicators: - Stable velocity - Managed technical debt - Measurable customer satisfaction - Positive team morale