

ИСП 1.3 Анализ АРМ и руководство по использованию Jira с автоматизацией установки ПО (Windows)

1. Автоматизированное рабочее место (АРМ) специалиста

1.1. Аппаратное обеспечение

- ПК/ноутбук: 4+ ядер CPU, 16–32 ГБ ОЗУ, SSD 512 ГБ+.
- Два монитора 24–27".
- Сеть: стабильное подключение 50 Мбит/с+, проводное предпочтительно.
- Периферия: гарнитура с шумоподавлением, веб-камера 1080p, ИБП.

1.2. Программное обеспечение

- ОС: Windows 10/11 (возможны macOS/Linux по политике).
- Браузер: Chrome / Edge / Firefox.
- Офис: MS Office / LibreOffice, PDF-просмотрщик.
- Коммуникации: Teams/Slack, корпоративная почта и календарь.
- Инструменты разработчика (по необходимости): Git, IDE, Docker Desktop, Postman.
- Безопасность: антивирус/EDR, VPN, шифрование, менеджер паролей, MFA/SSO.

1.3. Информационное обеспечение

- Репозитории знаний: Wiki (Confluence), шаблоны (устав, DoR/DoD, чек-листы).
- Хранилища кода/артефактов: GitHub/GitLab/Bitbucket, Nexus/Artifactory.
- Трекер задач: Jira (проекты, роли, статусы, метрики).
- Регламенты: политика доступа, SLA, схема разрешений в Jira.
- Интеграции: CI/CD, календарь релизов, отчёты.

2. Выбранное ПО: Jira (общее)

- Назначение: управление задачами, бэклогом, спринтами, Kanban-потоками, релизами и SLA.
- Развёртывание: Jira Cloud (браузер) или Jira Data Center (сервер).
- Модель данных: проекты → типы задач → поля → workflow → доски.
- Роли: администратор сайта, администратор проекта, участник, наблюдатель.

3. Руководство пользователя (Jira)

3.1. Создание проекта

1. Projects → Create project.
2. Шаблон: Scrum (со спринтами) или Kanban (непрерывный поток).
3. Имя, ключ проекта, участники.
4. Team-managed или Company-managed.

3.2. Структура задач

Иерархия: Epic → Story/Task → Sub-task; отдельный тип Bug для дефектов.

5. Create → Issue Type → заполните Summary, Description, Assignee, Priority, Labels.
6. Для Scrum укажите Story Points; добавьте связи (Link issue) и Epic Link.
7. Вложения/чек-боксы в описании.

3.3. Бэклог и спринты (Scrum)

8. Backlog: создайте эпики, распределите истории.
9. Оцените задачи (Story Points).
10. Создайте спринт, задайте Sprint Goal и длительность.
11. Start Sprint и ежедневные стендапы.

3.4. Kanban-поток

12. Колонки: To Do / In Progress / In Review / Done.
13. WIP-лимиты.
14. Swimlanes и Quick Filters.

3.5. Workflow и статусы

Базовый пример: To Do → In Progress → In Review → Done.

- Изменяйте статусы и переходы (conditions, validators, post-functions).
- Привязывайте схемы к типам задач (Workflow Scheme).

3.6. Роли и доступ

15. Project settings → People: роли Admin/Member/Viewer.
16. Permissions: права на просмотр/создание/редактирование/управление спринтами.
17. Security Level: ограничение видимости.
18. Components: подсистемы и Component Lead (автоприсвоение).

3.7. Отслеживание, коммуникация, документация

- Комментарии, @упоминания, вложения.
- Watchers.

- Roadmap: планы по эпикам и версиям.
- Документация: Confluence (Create linked page) или подробные описания в задачах.

3.8. Фильтры и JQL (примеры)

Все открытые задачи моей команды:

project = SP AND statusCategory != Done ORDER BY priority DESC, created DESC

Мои задачи в работе:

assignee = currentUser() AND status = "In Progress"

Просроченные задачи:

due <= now() AND statusCategory != Done

По компоненту и типу:

project = SP AND component = "Backend" AND issuetype in (Bug, Task)

3.9. Доски и отчёты

- Scrum: Burndown, Velocity, Sprint Report, Release burndown.
- Kanban: Cumulative Flow Diagram, Control Chart.
- Dashboards: Filter Results, Sprint Health, Created vs Resolved, Pie Chart.

3.10. Автоматизация (Automation)

- Issue Created → If Component=X → Assign to Lead.
- When PR merged → Transition issue.
- Ежедневная эскалация просроченных задач по JQL.
- Автоматическая постановка меток/полей.

3.11. Интеграции

- GitHub/GitLab/Bitbucket: PR/ветки, статусы CI.
- Мессенджер: уведомления в канал (webhook/приложение).
- Календарь релизов (плагины).
- Service Desk/Forms (порталы и SLA).

3.12. Импорт/экспорт и резервирование

- Импорт CSV с сопоставлением полей.
- Экспорт CSV/JSON.
- Резервное копирование по регламенту (Cloud/DC).

3.13. Шаблоны и соглашения

- Нейминг: [Модуль] Короткое действие (напр., [Auth] Добавить 2FA).
- Метки: team-backend, tech-debt, urgent, blocked.

- Definition of Ready и Definition of Done.
- Приоритеты: P1–P4 со SLA.

3.14. Горячие клавиши

Действие	Клавиши
Создать задачу	С
Поиск	/
Редактировать	Е
Назначить	М
Быстрое действие	.
Навигация по колонкам	[/]

4. Критерии готовности АРМ и проекта

- Пользователи добавлены, MFA/SSO включены.
- Проект создан, доска настроена (колонки, WIP).
- Workflow/статусы/права определены.
- Фильтры/JQL и дашборды готовы.
- Интеграции с репозиториями/мессенджером настроены.
- Шаблоны задач, DoR/DoD и бэклог сформированы.
- Регламент резервного копирования/экспорта согласован.

Приложение А. Чек-лист (Scrum)

19. Создать проект (Scrum, team-managed).
20. Добавить участников (Admin/Member/Viewer).
21. Настроить доску: колонки, swimlanes, quick filters.
22. Компоненты (Frontend/Backend/DevOps/QA) и версии (v1.0, v1.1).
23. Создать эпики, истории, оценить Story Points.
24. Подготовить спринт, задать Sprint Goal, запустить.
25. Включить отчёты и дашборды.
26. Создать правила автоматизации (автоприсвоение, эскалации).
27. Оформить DoR/DoD в Wiki и сослаться из эпиков.
28. Подписаться на рассылку «просроченные задачи».