# Agile Scrum Werkwijze — Project 1: Gusto Italiano

## Inleiding

Dit document beschrijft de Agile Scrum werkwijze voor het \*\*Gusto Italiano\*\* websiteproject. Het bevat teamrollen, ceremonies, artefacten, sprintplanning, backlogvoorbeelden, Definition of Done (DoD) en technische richtlijnen passend bij de gegeven technische eisen (React, Node.js/Express, MySQL/Postgres/MongoDB, JWT, Stripe).

---

## Projectoverzicht (samenvatting)

\*\*Klant:\*\* Marco Donatelli — Gusto Italiano (Antwerpen)

\*\*Doel:\*\* Een aantrekkelijke, gebruiksvriendelijke website waar klanten het menu kunnen bekijken, de locatie vinden, een account aanmaken en bestellingen plaatsen voor afhaling met online betalen via Stripe.

\*\*Technische eisen:\*\*

- Frontend: React

- Backend: Node.js + Express

- Database: MySQL / PostgreSQL / MongoDB (team kiest)

- Auth: JWT

- Betaalprovider: Stripe

---

## Team en rollen (aanbevolen)

- \*\*Product Owner (PO):\*\* Vertegenwoordigt Marco / stakeholder — prioriteert backlog, accepteert features.

- \*\*Scrum Master (SM):\*\* Faciliteert scrums, verwijdert obstakels, bewaakt proces.

- \*\*Development Team (3–5 devs):\*\* Full‑stack developers (React + Node). Team is cross‑functioneel.

- \*\*Tester/QA (kan rol rouleren):\*\* Schrijft testcases, voert acceptatie- en regressietests uit.

- \*\*Designer (optioneel):\*\* UI/UX, zorgt voor aantrekkelijke look conform Marco's visie.

\*\*Notitie:\*\* in een studententeam kunnen rollen rouleren; PO kan een teamlid zijn dat contact houdt met Marco.

---

## Scrum ceremonies & frequenties

- \*\*Sprintlengte:\*\* 2 weken (aanbevolen).

- \*\*Sprint Planning (2 uur max, start van sprint):\*\* PO presenteert top backlog items; team kiest wat haalbaar is; taken worden opgesplitst en geschat.

- \*\*Daily Stand-up (15 min):\*\* Kort: wat deed je gisteren, wat plan je vandaag, blokkades.

- \*\*Sprint Review (1 uur):\*\* Demo van afgeronde features voor PO / stakeholder.

- \*\*Sprint Retrospective (45–60 min):\*\* Wat ging goed, wat kan beter, concrete acties.

- \*\*Backlog Refinement (1u per week):\*\* Items verduidelijken en splitsen, inschattingen bijwerken.

---

## Artefacten

- \*\*Product Backlog:\*\* geprioriteerde lijst van user stories en technische taken.

- \*\*Sprint Backlog:\*\* items geselecteerd voor huidige sprint + taken en schattingen.

- \*\*Increment / Release:\*\* werkende software klaar voor demo of productie (minimaal bruikbaar).

- \*\*Definition of Done (DoD):\*\* criteria waaraan een backlog item moet voldoen om als 'klaar' te gelden.

---

## Definition of Done (standaard)

Een backlog item is \*Done\* wanneer:

- Code is geschreven en gecommit in feature branch.

- Unit tests en integratietests (backend) toegevoegd waar relevant en draaien groen.

- E2E of integratietest waar passend (minimaal 1 covered flow).

- Code review is uitgevoerd en opmerkingen verwerkt.

- Feature is gedocumenteerd (README of API doc).

- Build slaagt in CI (lint + tests).

- Feature is gedeployed naar staging (optioneel) of lokaal reproduceerbaar.

- Design en toegankelijkheid (basis WCAG checks) gecontroleerd.

- PO heeft feature geaccepteerd tijdens sprint review.

---

## Voorstel Sprintindeling (8 weken — 4 sprints)

\*\*Sprint 0 — Voorbereiding (optioneel, 1 week)\*\*

- Projectsetup: repo, branch-strategie, basis CI, dev environment, keuze DB.

- Wireframes en basis UI-stijl afspreken (kleur, logo, fonts).

- Basis README en run-instructies.

\*\*Sprint 1 (week 1–2) — Auth & basis UI\*\*

- User registration & login (JWT).

- Basis layout: header, footer, homepage, menu page skeleton.

- Datamodel users & sessions.

- CI lint & test pipeline.

\*\*Sprint 2 (week 3–4) — Menu & locatie\*\*

- CRUD menu items (admin) + publieke menu weergave.

- Locatie pagina met kaart en openingstijden.

- Frontend styling en responsive basis.

\*\*Sprint 3 (week 5–6) — Bestellen & afhaling\*\*

- Winkelwagen flow (client-side).

- Checkout flow: order aanmaken op backend, keuze 'afhaling' + tijdslot.

- Integratie met Stripe (testmodus) voor betalingen.

- Order history voor gebruiker.

\*\*Sprint 4 (week 7–8) — Polish, tests & release\*\*

- E2E tests (registratie → order → betaling).

- UI polish, performance, accessibility fixes.

- Deployment naar productie (Heroku/Render/Railway).

- Einddemo en oplevering rapport.

---

## Product Backlog (voorbeelden — geprioriteerd)

Hieronder 20 voorbeeld user stories & technische tasks. Prioriteit: Hoog → Laag.

### Hoog

1. \*\*US-01 — Account aanmaken\*\*

- Als klant wil ik een account kunnen aanmaken zodat mijn naam & adres opgeslagen worden.

- Acceptatie: POST /auth/register werkt, wachtwoord gehashed, validatie email.

2. \*\*US-02 — Inloggen / JWT sessie\*\*

- Als klant wil ik kunnen inloggen zodat ik bestellingen kan plaatsen en mijn gegevens beheren.

- Acceptatie: POST /auth/login retourneert JWT, protected endpoints vereisen token.

3. \*\*US-03 — Menu bekijken (publiek)\*\*

- Als bezoeker wil ik het huidige menu kunnen bekijken met prijzen en beschrijving.

- Acceptatie: publieke GET /menu endpoint, responsive UI.

4. \*\*US-04 — Bestelling plaatsen (afhaling)\*\*

- Als klant wil ik producten in winkelwagen plaatsen en een bestelling plaatsen voor afhaling.

- Acceptatie: order opgeslagen met user\_id, items, gekozen tijdslot en status 'pending'.

5. \*\*US-05 — Online betalen via Stripe\*\*

- Als klant wil ik veilig online betalen zodat ik gemakkelijk mijn bestelling betaal.

- Acceptatie: Stripe testintegratie, betaling bevestigt order status 'paid'.

6. \*\*US-06 — Order history & status\*\*

- Als klant wil ik mijn eerdere bestellingen zien en status volgen.

- Acceptatie: GET /orders voor ingelogde gebruiker.

### Midden

7. \*\*US-07 — Locatie & kaart\*\* — Google Maps / Leaflet integratie.

8. \*\*US-08 — Admin: menu items beheren\*\* (CRUD).

9. \*\*US-09 — Time-slot selectie & conflictcontrole\*\*.

10. \*\*US-10 — Profielpagina: adres bewerken\*\*.

### Laag

11. \*\*US-11 — Reviews / ratings (optioneel)\*\*.

12. \*\*US-12 — Filters op menu: soorten, allergenen\*\*.

13. \*\*US-13 — E-mail bevestigingen voor orders (optie)\*\*.

14. \*\*US-14 — Export orders als CSV (admin)\*\*.

### Technische taken

- \*\*T-01:\*\* Setup DB (schema voor users, menu\_items, orders, order\_items).

- \*\*T-02:\*\* CI pipeline (lint, tests).

- \*\*T-03:\*\* SSL / env management voor Stripe keys.

- \*\*T-04:\*\* Basic input validation & sanitization.

---

## Voorbeeld user story + taken (US-04 Bestelling plaatsen)

\*\*User story:\*\* Als klant wil ik producten in winkelwagen plaatsen en een bestelling plaatsen voor afhaling.

\*\*Acceptatiecriteria:\*\*

- Kan producten toevoegen/verwijderen in winkelwagen.

- Checkout creëert order met correcte itemgegevens en totaalprijs.

- Order heeft afhaaltijd en status 'pending'.

- Alleen ingelogde gebruikers kunnen een bestelling plaatsen (of guest checkout indien gekozen).

\*\*Taken:\*\*

1. Frontend: winkelwagen component + UI (React state / context).

2. Backend: POST /orders endpoint (valideer token, bereken totaal, sla order + items op).

3. DB: orders + order\_items tables/collections.

4. Integratie: redirect naar Stripe Checkout (test) of PaymentIntent flow.

5. Tests: unit test for order calculation; E2E checkout flow.

---

## Acceptatie & Testing

- \*\*Unit tests:\*\* backend (business logic) + frontend (utility functions, components).

- \*\*Integratietests:\*\* auth flow + order creation.

- \*\*E2E tests:\*\* volledige gebruikersflow (Cypress/Playwright): register → add to cart → checkout → payment success simulation.

- \*\*QA checklist:\*\* security (JWT expiration), input validation, XSS prevention, error handling, mobile responsiveness.

---

## Tech- & Securityrichtlijnen

- \*\*Database keuze:\*\* kies PostgreSQL voor relationele consistency; MongoDB is OK als team comfortabel is met documentmodel.

- \*\*Auth:\*\* JWT met refresh tokens (indien nodig). Bewaar niet gevoelige data in JWT, valideer server-side.

- \*\*Passwords:\*\* bcrypt hashing (min cost 10).

- \*\*Stripe:\*\* gebruik test keys in dev, set webhook endpoint veilig in production; verwerk webhooks idempotent.

- \*\*Env management:\*\* .env voor secrets; zorg dat secrets niet in repo staan.

- \*\*CORS:\*\* configureer alleen benodigde origins.

- \*\*Backup:\*\* eenvoudige DB dump/backup bij productie.

---

## Definition of Ready (DoR) — wanneer kan een item in sprintplanning?

- User story heeft duidelijke beschrijving en acceptatiecriteria.

- Story is geschat (story points).

- Afhankelijkheden zijn duidelijk of opgevoerd als taken.

- UI designs of mockups beschikbaar indien relevant.

---

## Schatting & Velocity (advies)

- Gebruik \*\*story points (Fibonacci 1,2,3,5,8)\*\* voor inschatting.

- Start velocity: team kan in sprint 1 conservatief 10–15 story points plannen (afhankelijk van teamgrootte/ervaring).

- Houd velocity per sprint bij en pas planning aan.

---

## Voorbeeld sprint backlog (Sprint 1)

| Taak | Eigenaar | Schatting (SP) | Status |

|------|----------|----------------:|--------|

| Repo init + README | Dev A | 1 | Done |

| Auth endpoints (register/login) | Dev B | 3 | In Progress |

| Frontend: login/register UI | Dev C | 3 | Todo |

| DB schema users | Dev B | 2 | Todo |

| CI: lint + test job | Dev A | 2 | Todo |

---

## Templates & Artefacten (voor teamgebruik)

- \*\*Sprint board kolommen:\*\* To Do | In Progress | In Review | Done

- \*\*Pull request template:\*\* beschrijving, changes, tests gedaan, how to test, screenshots.

- \*\*Issue template:\*\* omschrijving, stappen reproducible, verwachte vs werkelijke, omgeving.

---

## Deployment & Devops (kort)

- \*\*Staging:\*\* Deploy naar Render/Railway voor demo tijdens sprint review.

- \*\*Production:\*\* Heroku/Render/Vercel (frontend) + managed DB (Heroku Postgres / Render Postgres).

- \*\*CI:\*\* GitHub Actions — run lint, run tests, build. Deploy on merge to main (with manual approval).

---

## Communicatie & Stakeholdermanagement

- PO (teamlid) plant wekelijkse check-in met Marco (15–30 min) voor feedback en prioriteit.

- Gebruik Slack / MS Teams voor dagelijkse communicatie + GitHub Issues & Project board.

- Verzamel feedback tijdens sprint review en verwerk in backlog.

---

## Bijlagen / aanvullend materiaal (suggesties)

- Wireframes (Figma links)

- API contract (OpenAPI spec)

- Database ERD / schema

---

\*\*Einde document\*\*