

Module 9 — Bonus : Claude Craft

Jour 2 — Demo 30min

The Bearded Bear

Février 2026



Table des matières

Module 9 : Bonus – Claude Craft (30min)	2
Objectifs	2
1. Qu'est-ce que Claude-Craft ?	2
Definition	2
Analogie	2
Ce que Claude-Craft ajoute	2
2. Installation	3
Commande d'installation	3
Technologies supportees	3
Ce qui est installe	3
3. Les 63 Agents Specialises	4
Categories d'agents	4
Utilisation d'un agent	5
Exemple : @tdd-coach	5
Exemple : @react-reviewer	5
4. Les 204 Commandes (26 Namespaces)	5
Namespaces principaux	5
Commandes specifiques par technologie	6
Demonstration rapide	6
5. BMAD v6 Framework	6
Les 3 Tracks	6
Quality Gates	7
Workflow typique	7
6. Ralph Wiggum	7
Concept	7
Utilisation	7
Validators de Definition of Done (DoD)	7
7. QA Recette	8
Concept	8
Prerequis	8
Utilisation	8
Regle d'or	8
8. Pour Aller Plus Loin	9
Ressources	9
Prochaines etapes	9
Points Cles a Retenir	9

Module 9 : Bonus – Claude Craft (30min)

Objectifs

A la fin de ce module, vous aurez : - Compris ce qu'est Claude-Craft et ce qu'il apporte par rapport a Claude Code natif - Vu une demonstration d'installation et d'utilisation - Un aperçu des agents, commandes et workflows disponibles - Les ressources pour approfondir si vous souhaitez l'adopter

Note : Ce module est un **teaser/demo**, pas une formation complete. Pour une formation approfondie sur Claude-Craft, consultez la formation dediee.

1. Qu'est-ce que Claude-Craft ?

Definition

Claude-Craft est un framework d'extension pour Claude Code developpe par The Bearded CTO. Il ajoute a Claude Code natif : - Des **agents specialises** par technologie et par role - Des **commandes structurees** pour des taches recurrentes - Un **framework de gestion de projet** (BMAD v6) - Des **workflows automatises** pour le developpement complet

Analogie

Claude Code = Un terminal puissant avec un assistant IA
Claude-Craft = Un framework complet qui structure cet assistant
avec des agents, des commandes et des workflows

Ce que Claude-Craft ajoute

Aspect	Claude Code natif	Avec Claude-Craft
Agents	Task tool generique	63 agents specialises par role
Commandes	Slash commands custom	204 commandes preconfigurees
Architecture	Regles dans CLAUDE.md	Clean Architecture enforced
Qualite	Audit manuel	Audit automatise multi-dimensions
Projet	Gestion libre	BMAD v6 avec quality gates
CI/CD	Configuration manuelle	Templates et workflows prêts

2. Installation

Commande d'installation

```
# Installation avec NPX (recommande)
npx @the-bearded-bear/claude-craft install . --tech=react --lang=fr

# Parametres :
# .            -> Repertoire cible (. = projet courant)
# --tech       -> Technologie (react, symfony, flutter, angular, etc.)
# --lang       -> Langue (fr, en, es, de, pt)
```

Technologies supportees

Stack	Version	Architecture
.NET / C#	10 LTS / C# 14	Clean Architecture
Symfony / PHP	8.0 / PHP 8.5	Clean Architecture
Flutter / Dart	3.38 / Dart 3.10	Clean Architecture
React	19.x	Feature-based
React Native	0.76+	Feature-based
Angular	19.x	Domain-driven
Vue.js	3.5+	Composition API
Laravel	12.x / PHP 8.5	Clean Architecture
Python	3.13+	Clean Architecture
PHP	8.5	Clean Architecture

Ce qui est installe

```
.claude/
  CLAUDE.md           # Instructions principales
  settings.json        # Configuration + hooks
  rules/              # 12 fichiers de regles
    01-workflow-analysis.md
    04-solid-principles.md
    05-kiss-dry-yagni.md
    07-testing.md
    09-git-workflow.md
    11-security.md
    12-context-management.md
```

```

...
references/           # Documentation technique spécifique
  react/              # (ou symfony/, flutter/, etc.)
commands/            # Slash commands
agents/              # Definitions d'agents
skills/              # Competences a la demande

```

3. Les 63 Agents Specialises

Catégories d'agents

Categorie	Agents	Nombre
Common	api-designer, database-architect, devops-engineer, performance-auditor, refactoring-specialist, tdd-coach, uiux-orchestrator, ui-designer, ux-ergonome, accessibility-expert, research-assistant, ralph-conductor	12
Tech Reviewers	symfony-reviewer, flutter-reviewer, react-reviewer, python-reviewer, angular-reviewer, laravel-reviewer, vuejs-reviewer, reactnative-reviewer, csharp-reviewer, php-reviewer	10
Infrastructure	docker (5), coolify (4), kubernetes (5), opentofu (5), ansible (5), hcloud (5), pgbouncer (5), frankenphp (5)	39

Categorie	Agents	Nombre
Project	product-owner, tech-lead	2

Utilisation d'un agent

Dans Claude Code, invoquer un agent :

@tdd-coach Guide-moi pour ecrire les tests de mon service de paiement

@react-reviewer Revois ce composant React et verifie les bonnes pratiques

@database-architect Propose un schema pour un systeme de notifications

@devops-engineer Configure le pipeline CI/CD pour ce projet

Exemple : @tdd-coach

L'agent TDD Coach guide le developpeur a travers le cycle Red-Green-Refactor avec des instructions specifiques a la technologie du projet.

Exemple : @react-reviewer

L'agent React Reviewer effectue une code review specifique React : hooks, re-renders, accessibilite, patterns recommandes.

4. Les 204 Commandes (26 Namespaces)

Namespaces principaux

Namespace	Commandes cles	Nombre
/common:	pre-commit-check, ralph-run, setup-project-context	13
/workflow:	init, analyze, plan, design, implement, status	9
/team:	audit, sprint, security, delivery	4
/qa:	recette, fix, tdd, regression, report	6
/sprint:	next-story, transition, status, dev	5
/gate:	validate-prd, validate-story, validate-backlog	7
/project:	run-sprint, run-epic, board, trace, checkpoint	34

Commandes spécifiques par technologie

Chaque technologie a son propre namespace avec des commandes adaptées :

```
/react:generate-component    # Genere un composant React
/react:check-architecture    # Verifie l'architecture React
/react:accessibility-check    # Audit accessibilite

/symfony:generate-crud        # Genere un CRUD Symfony
/symfony:check-compliance    # Audit global Symfony

/python:generate-endpoint     # Genere un endpoint FastAPI
/python:async-check           # Verifie les patterns async
```

Démonstration rapide

```
# Initialiser un workflow
/workflow:init

# Lancer un audit complet avec une équipe d'agents
/team:audit --sequential

# Générer un composant
/react:generate-component Button --variant=primary
```

5. BMAD v6 Framework

Les 3 Tracks

BMAD (Business-Mission-Architecture-Delivery) est un framework de gestion de projet intégré à Claude-Craft :

Track	Setup	Phases	Usage
Quick Flow	< 5 min	Implement uniquement	Bug fixes, hotfixes
Standard	< 15 min	Plan -> Design -> Implement	Nouvelles features
Enterprise	< 30 min	Analyze -> Plan -> Design -> Implement	Plateformes, projets majeurs

Quality Gates

Chaque phase doit passer des gates de qualite avant de continuer :

Gate	Seuil	Verification
PRD (Product Requirements)	$\geq 80\%$	Completude des requirements
Tech Spec	$\geq 90\%$	Precision de la specification technique
INVEST	6/6	Qualite des user stories
Sprint Ready	100%	Preparation du sprint
Story DoD	100%	Definition of Done
Spec Alignment	$\geq 85\%$	Alignement code/spec

Workflow typique

1. `/workflow:init` -> Choisir le track (Quick/Standard/Enterprise)
2. `/workflow:analyze` -> Analyse du besoin (Enterprise uniquement)
3. `/workflow:plan` -> Planification technique
4. `/workflow:design` -> Conception detaillee
5. `/workflow:implement` -> Implementation avec TDD
6. `/workflow:status` -> Suivi de l'avancement

6. Ralph Wiggum

Concept

Ralph Wiggum est une boucle IA continue qui fait tourner Claude Code jusqu'à ce qu'une tache soit completement terminee. Il gere automatiquement les erreurs, les relances et les validations.

Utilisation

```
/common:ralph-run "Implemente le systeme de notifications avec tests et  
↳ documentation"
```

Validators de Definition of Done (DoD)

Ralph valide l'achevement de la tache avec differents types de validateurs :

Validator	Description	Exemple
command	Commande qui doit reussir (exit 0)	<code>npm test</code>
output_contains	La sortie doit contenir un texte	"All tests passed"
file_changed	Un fichier doit avoir ete modifie	<code>src/notification.ts</code>
hook	Un hook personnalise de validation	Script de verification
human	Validation humaine requise	Approbation manuelle

7. QA Recette

Concept

QA Recette est un systeme de tests d'acceptance automatises qui utilise un navigateur Chrome pour valider les fonctionnalites du point de vue utilisateur.

Prerequis

- Extension Chrome Claude v1.0.36+
- Claude Code avec `--chrome` ou `/chrome`

Utilisation

```
# Tester une user story
/qa:recette --scope=story --id=US-001

# Tester un sprint complet
/qa:recette --scope=sprint --id=Sprint-3

# Reprendre une session interrompue
/qa:recette --resume=REC-20260130-143022
```

Regle d'or

Un bug corrige ne doit JAMAIS reapparaître. Chaque correction est accompagnee d'un test de regression automatique.

8. Pour Aller Plus Loin

Ressources

Ressource	Description
Formation Claude-Craft	Formation dediee de 2 jours
Documentation	docs/QUICKSTART.md
CLI Reference	docs/CLI-REFERENCE.md
Liste des commandes	docs/COMMANDS.md
Liste des agents	docs/AGENTS.md
FAQ	docs/FAQ.md

Prochaines etapes

Si Claude-Craft vous interesse :

1. **Installer** : `npx @the-bearded-bear/claude-craft install . --tech=<votre-stack> --lang=fr`
2. **Explorer** : Lancer `/workflow: init` et suivre le guide
3. **Approfondir** : Suivre la formation dediee Claude-Craft

Points Cles a Retenir

1. **Claude-Craft** etend Claude Code avec des agents, commandes et workflows structures
2. **63 agents** specialises couvrent tous les roles du developpement
3. **204 commandes** dans 26 namespaces pour des taches recurrentes
4. **BMAD v6** structure la gestion de projet avec des quality gates
5. **Ralph Wiggum** automatise les taches jusqu'a completion
6. **QA Recette** valide les fonctionnalites dans un vrai navigateur
7. **Ce module est un apercu** : la formation dediee Claude-Craft couvre ces sujets en profondeur

Duree : 30min **Prochain module** : Module 10 : Atelier Final