



Software Engineering in Industrial Practice (SEIP)

Dr. Ralf S. Engelschall

Custom Software Development
CSD

Commercial development of **non-standardised, fully individualised, and non-reusable company-specific** software for a **single** customer.


Standard Software Development
STD

Commercial development of standardised, **partially customisable, and fully reusable domain-specific** software for **many** customers.


Open Source Software Development
OSS

Non-commercial development of standardised, **highly customisable, and fully reusable generic** software for **many** customers.


Class: Graphics & Media

target audience: consumers & enterprises

Graphics Editing Application
GEA

Software for editing and rendering graphics in vector and bitmap format.



Examples: Cinema4D, Maya, Blender, After Effects, Illustrator, Inkscape, Scribus, Photoshop, GIMP, etc.

Graphics Animation Engine
GAE

Software for animating the 2D/3D virtual worlds of games and overlays of TV productions.



Examples: Unity, Unreal Engine, CryENGINE, Godot, HUDS, SPX-GC, Holographics, H2R Graphics, etc.

Audio/Video-Processing System
AVS

Software for live-processing and post-production of audio/video based multimedia streams.



Examples: vMix, OBS Studio, VLC, Lossless Cut, Handbrake, Adobe Premiere, FFmpeg, Nimble, etc.

Class: Business & Data

target audience: consumers & enterprises

Office Productivity Application
OPA

Software for productivity in the desktop-based office environment.



Examples: PowerPoint, Excel, Word, Visio, OmniGraffle, LibreOffice, Outlook, XMind, Firefox, Chrome, etc.

Data Management System
DMS

Software for protocol-based storing and retrieving of persistent data.



Examples: NextCloud, PostgreSQL, CockroachDB, Redis, InfluxDB, Neo4j, Tendermind, Gitea, Vault, etc.

Class: Machinery & Network
Class: Machinery & Network

target audience: consumers & enterprises

Technical Control System
TCS

Software for controlling a physical machinery or technical system.



Examples: AquaTherm, AVM! FritzBox Firmware, BirdDog Camera Firmware, etc.

Business Information System
BIS

Software for driving business processes through interactive information management.



Examples: Vote, CampS, Mission Control, IPW, KEZ-PSC, TimeSheet, SAP ERP, OpenProject, etc.

Network Communication System
NCS

Software for protocol-based communication of data over a computer network.



Examples: Apache, NGINX, HAProxy, Mosquitto, RabbitMQ, Node-RED, Keycloak, etc.

Operating System Kernel
OSK

Software kernel for low-level operating a physical or virtual device and run programs on it.



Examples: Windows, macOS, iOS, Linux, FreeBSD, QNX, ChibiOS/RT, Kubernetes, Wildfly, etc.

Operating System Tools
OST

Software tools for high-level operating a physical or virtual computing device.



Examples: Coreutils, Bash, Vim, TMux, FZF, cURL, RSYNC, OpenSSH, etc.

Development Approaches

Software Prototyping SP

Develop an early sample or model of a software solution by mocking and cheating in order to just once test a concept, idea or process.

Example: Customer Sales Demo

Software Bricolage SB

Develop a single instance of a software solution by tinkering, cobbling and integrating partial solutions in order to prove feasibility or just provide a service.

Example: Company-Internal SaaS

Software Craftsmanship SC

Develop a production-grade software solution by professional, clean but plain craftsmanship means in order to solve a usually complicated problem.

Example: Open Source Framework

Software Engineering SE

Develop a production-grade software solution by a professional, risk-hedged engineering approach in order to solve a usually complex problem.

Example: Business Information System

Development Approaches: Characteristics Comparison *

	Effort: Person-Days	Effort: Persons	Process: Risk-Hedge	Process: Traceability	Solution: Target Technology	Solution: Production-Grade	Solution: Sustainability	Solution: Claim	Solution: Life-Time Months	Solution: Lines of Code (K)
Software Prototyping	1-20	1-2	-	-	-	-	-	5%	0-3	0-3
Software Bricolage	5-100	1-2	-	-	X	(x)	-	60%	3-24	1-10
Software Craftsmanship	5-100	1-2	-	-	X	X	X	100%	24-48	5-25
Software Engineering	>150	5-50	X	X	X	X	X	80%	>48	>25

* All figures are just rough orders of magnitude for indication and illustration purposes.

Key Message

All four approaches are equally essential in practice. Which one(s) to choose, entirely depends on the particular requirements.

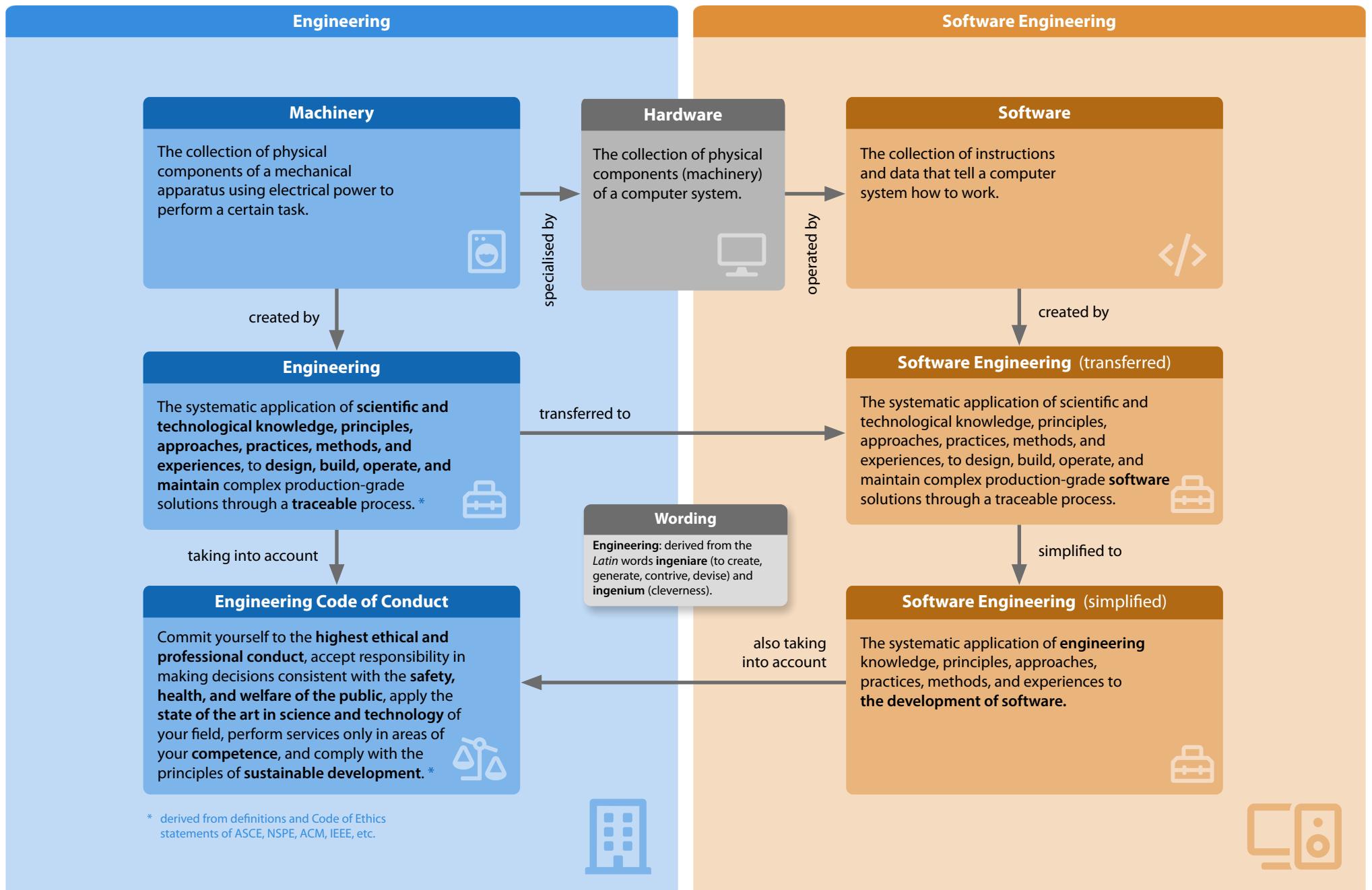
Development Approaches: Success Patterns

	Software Prototyping	Software Bricolage	Software Craftsmanship	Software Engineering
Performance Responsibility Model	One-Man-Show Single Mental	One-Man-Show Single Mental	One-Man-Show Single Mental/Documented	Team Play Separated Documented
Decisions Process Optimisation	Implicit Minimized Time	Implicit Partial Efficiency	Implicit/Explicit Partial Effectiveness	Explicit Complete Economics
Risks Stakeholders Mastering	Ignore Ignore Time-Constraint	Ignore Ignore Complexity	Ignore Ignore Complication	Mitigate Manage Complexity
Solutions Standards Efforts	Use Full Use Configuration	Use Partial Use Integration	Use Partial Potentially Create Programming	Use Partial Use Programming
Target Sustainability Traceability	Demo No No	Solution Partial No	Product Full Partial	Product Full Full

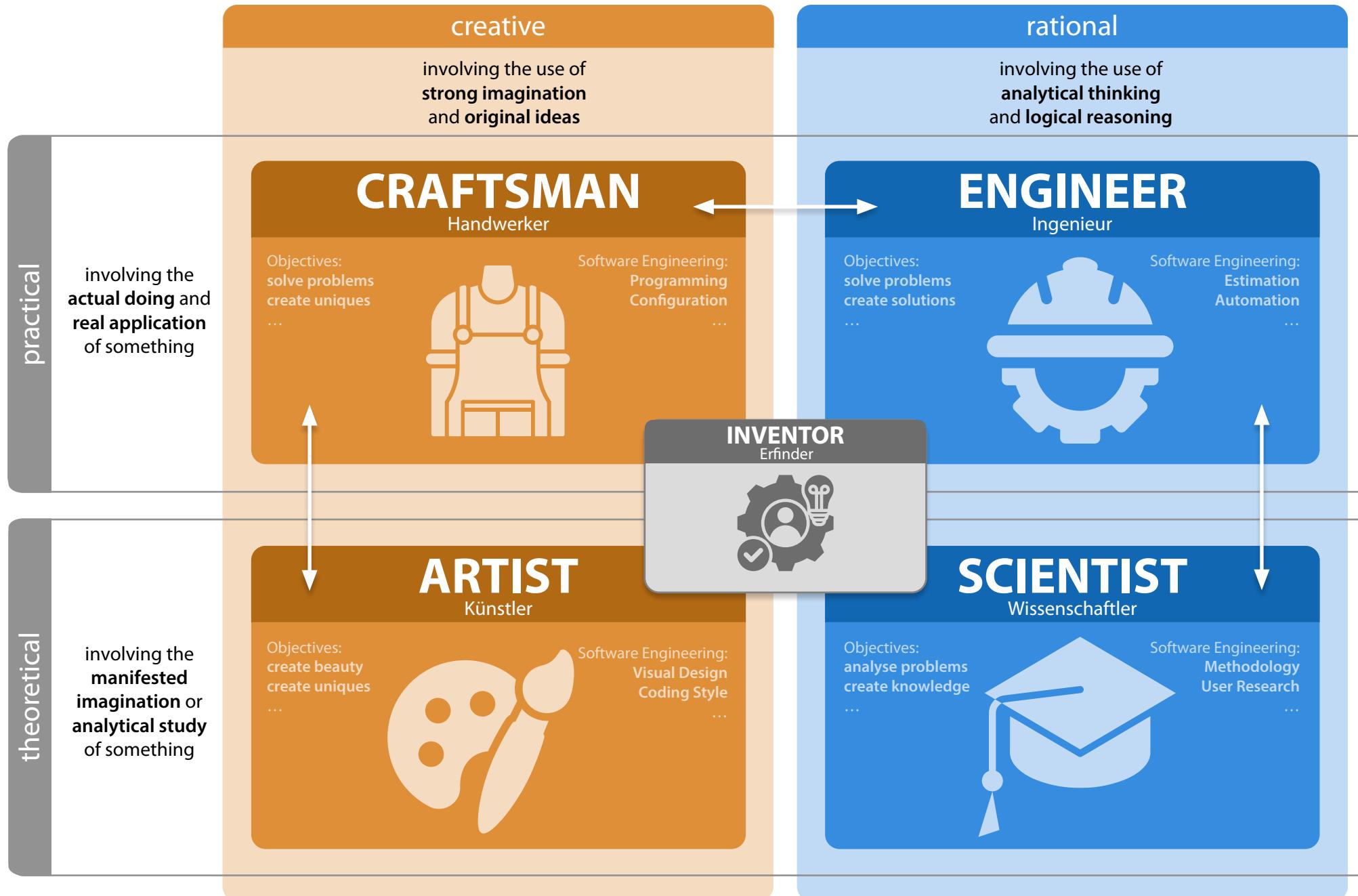
goal & approach

EF 01.2

Intellectual Content: Version 1.04 (2022-11-10). Authored 2021-2022 by Dr. Ralf S. Engelhardt (http://engelhardt.com). All Rights Reserved. Unauthorized Reproduction Prohibited. Licensed to Technische Universität München (TUM) for reproduction in Computer Science lecture contexts only.

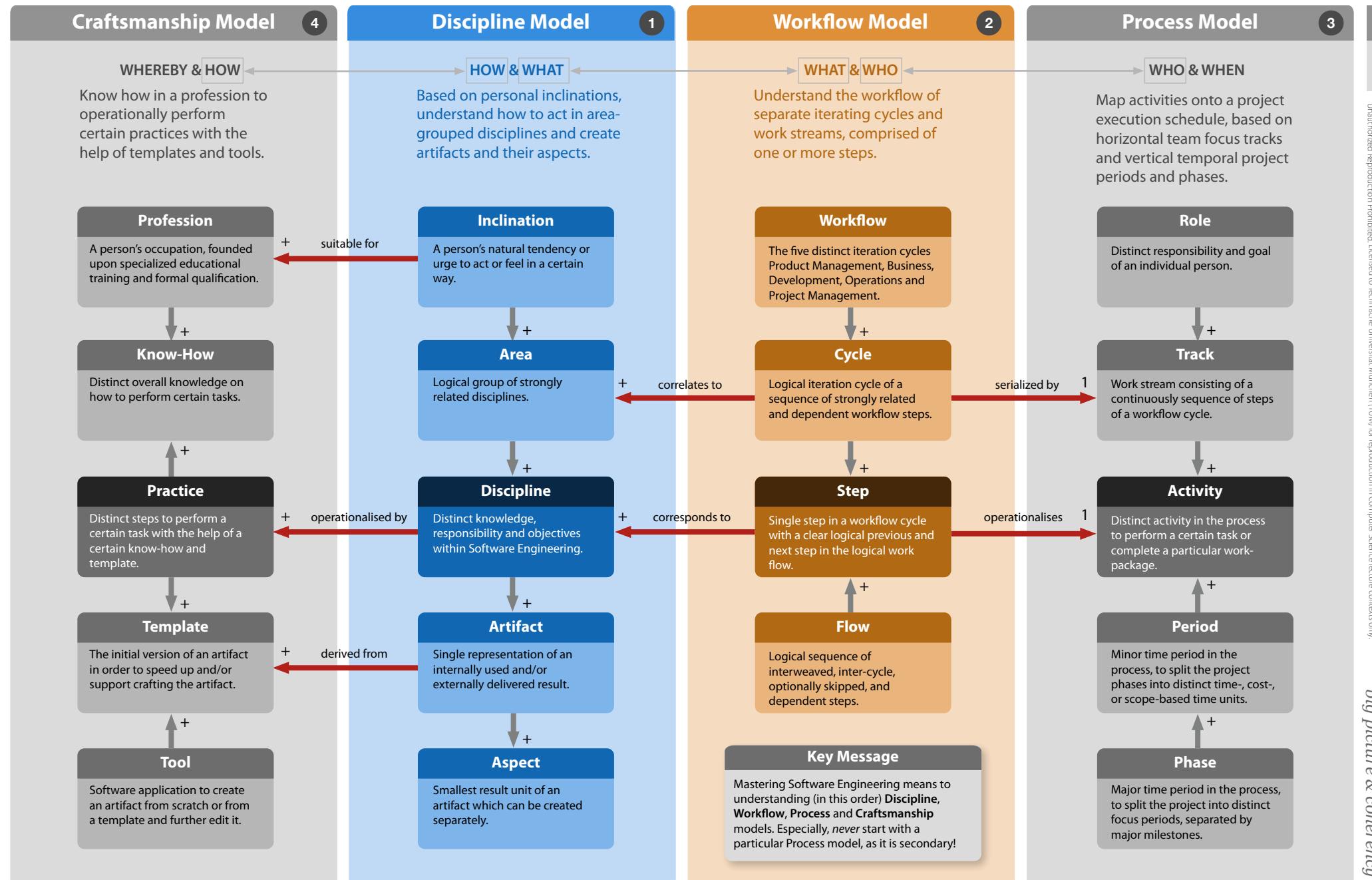


Profession Characteristics



Discipline Claim

FA FARSIGHTED weitblickend	TE TENET-ORIENTED grundsatzorientiert	TH THOUGHTFUL wohlüberlegt
<p>Be farsighted in your solution finding.</p> <p>Sei weitblickend in deiner Lösungsfindung.</p> <p>AR: Scalable Hub'n'Spoke DV: Plugin SPI</p>	<p>Oriентate yourself on fixed tenets in your approach and solution finding.</p> <p>Orientiere dich an festen Grundsätzen in deinem Vorgehen und deiner Lösungsfindung.</p> <p>AR: Separation of Concern DV: Strict Coding-Style</p>	<p>Act thoughtful in your approach and solution finding.</p> <p>Agiere wohlüberlegt in deinem Vorgehen und deiner Lösungsfindung.</p> <p>AR: Modularization DV: Algorithmical Control Structure</p>
HO HOLISTICALLY ganzheitlich	AD ADEQUATE angemessen	FE FEASIBLE machbar
<p>Think holistically and in the long-term when finding your solutions.</p> <p>Denke ganzheitlich und langfristig in deiner Lösungsfindung.</p> <p>AR: Walking Skeleton Design DV: Consistent Error Handling</p>	<p>Ensure that your approach and solutions are adequate to the boundary conditions.</p> <p>Sorge dafür, daß dein Vorgehen und deine Lösungen angemessen zu den Rahmenbedingungen sind.</p> <p>AR: No Cloud-Native Complexity DV: No Over-Engineered Abstractions</p>	<p>Ensure that your approach and solutions can be realised at reasonable costs.</p> <p>Sorge dafür, daß dein Vorgehen und deine Lösungen mit vernünftigen Kosten realisiert werden können.</p> <p>AR: Existing Framework Functionality DV: Realistic Programming Model</p>
IN INCREMENTAL inkrementell	VA VALUEABLE wertvoll	SU SUSTAINABLE nachhaltig
<p>Apply the depth of your discipline incrementally.</p> <p>Wende die Tiefe deiner Disziplin inkrementell an.</p> <p>AR: Identified Solution Cruxes DV: Minimum Viable Product</p>	<p>Provide clearly recognizable added values with your approach and solutions.</p> <p>Liefere klar ersichtliche Mehrwerte mit deinem Vorgehen und deinen Lösungen.</p> <p>AR: Technology Stack Design DV: User-Story-Driven Functionality</p>	<p>Create sustainable solutions that are well integrated into their environment.</p> <p>Erschaffe nachhaltige Lösungen, die gut in ihre Umgebung integriert sind.</p> <p>AR: Interoperable Interfaces DV: Maintainable Code</p>



Software Engineering Disciplines

ANALYSIS	AN
Software Requirements	REQ
Identify Needs: We understand which outcomes of the solution are most valuable to users.	3 BB
Requirements Engineer / Business Analyst	

 business-oriented & domain-specific

ARCHITECTURE	AR
Software Architecture	SWA
Design Software: We design an orthogonal, well-balanced and well-considered solution.	1 WB
Software Architect	

 constructive & technological

CONFIGURATION	CF
Software Versioning	VER
Version Artifacts: We place every artifact of the solution under strict version control.	1 BB
Configuration Manager	

 infrastructural & technological

ANALYTICS	AC
Software Reviewing	REV
Review Code: We regularly and semantically peer-review the source code of the solution.	4 WB
Software Tester	

 analytical & domain-specific

MANAGEMENT	MG
Product Management	PRD
Push Product: We continuously push the development and release of the solution to the users.	3 BB
Product Manager / Product Owner	

 people-oriented & process-oriented

EXPERIENCE	EX
User Experience	UXP
Optimize Workflows: We align the solution to the perspective of the target audience.	3 BB
User Experience Expert	

WB white-box view (details before whole)

DEVELOPMENT	DV
Software Development	DEV
Implement Code: We develop the solution outside-in, from coarse to fine aspects.	1 WB
Software Engineer / Software Developer	

BB black-box view (whole before details)

DELIVERY	DL
Software Deployment	DPL
Deploy Artifacts: We ship and deploy the solution through an automated and repeatable mechanism.	1 BB
System Engineer	

BB scalability layer (from 4/most to 1/least dispensable)

COMPREHENSION	CP
Usage Documentation	DOC
Document Solution: We adequately document the usage and operation of the solution.	2 WB
Technical Writer	

WB white-box view (details before whole)

ADJUSTMENT	AD
Project Coaching	COA
Support Members: We ensure that project members use state-of-the-art methodology, technology, and tools.	4 BB
Project Coach / Methodology Master	

BB black-box view (whole before details)