

DOMAIN DRIVEN DESIGN

DDD IS A SET OF DESIGN PRINCIPLES AND PATTERNS THAT HELP CREATE ELEGANT AND SCALABLE SW SYSTEMS IN COMPLEX DOMAINS. IT'S BASED ON A MODEL ORIENTED DESIGN APPROACH, WHICH ALLOWS TO ADDRESS COMPLEXITY BY CREATING FOCUSED ABSTRACTIONS. THE AIM IS TO ACCELERATE SW PROJECTS, ESPECIALLY IN AREAS CHARACTERIZED BY ORGANIZATIONAL OR OPERATIONAL COMPLEXITY.

DOMAIN: IT'S THE SCOPE OF KNOWLEDGE OR ACTIVITY OF AN ORGANIZATION.

MODEL: A SIMPLIFIED REPRESENTATION OF A DOMAIN, DISTILLING RELEVANT KNOWLEDGE, RULES, AND CHOICES WHILE IGNORING UNNECESSARY DETAILS.

PRINCIPLES OF DDD

UBIQUITOUS LANGUAGE: A SHARED LANGUAGE BETWEEN THE TECHNICAL TEAM AND DOMAIN EXPERTS, BASED ON THE CONCEPTUAL MODEL. IT'S USED TO ALIGN COMMUNICATION, DOCUMENTS AND CODE.

BOUNDED CONTEXT: IT DEFINES AN OPERATIONAL BOUNDARY WHERE A PARTICULAR MODEL IS WELL DEFINED AND APPLICABLE.

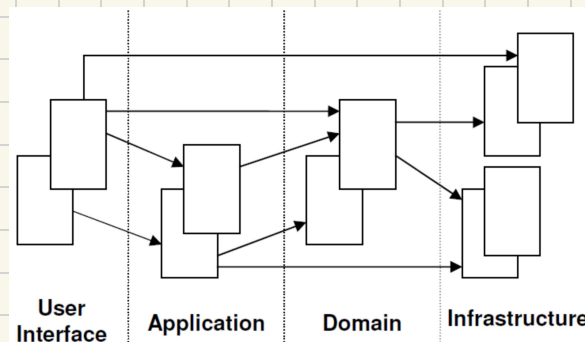
FOCUS ON CORE DOMAIN: GIVE PRIORITY TO THE MOST CRITICAL PART OF THE DOMAIN OVER GENERIC OR SECONDARY ASPECTS.

ITERATIVE AND AGILE PROCESSES

DDD EMPHASIZES CONTINUOUS DOMAIN LEARNING DURING DEVELOPMENT. AN ANALYSIS THAT IS TOO DETAILED AT THE BEGINNING RISKS BLOCKING THE DESIGN ON INCORRECT ASSUMPTIONS.

MODELS AND IMPLEMENTATIONS ARE CONSTANTLY REFINED BASED ON FEEDBACK AND NEW KNOWLEDGE GAINED, FOLLOWING AN AGILE APPROACH.

LAYERED ARCHITECTURE



USER INTERFACE: MANAGES INTERACTION WITH THE USER, INTERPRETING COMMANDS AND SHOWING DATA.

APPLICATION: COORDINATES APPLICATION ACTIVITIES, WITHOUT CONTAINING BUSINESS LOGIC.

DOMAIN: CONTAINS BUSINESS LOGIC AND STATE OF DOMAIN OBJECTS. IT'S THE ❤️ OF THE SW.

INFRASTRUCTURE: IT SUPPORTS THE OTHER LAYERS BY PROVIDING PERSISTENCE, COMMUNICATION, AND OTHER SUPPORT FEATURES.

MODEL COMPONENTS

ENTITIES. OBJECTS THAT ARE UNIQUELY IDENTIFIABLE, EVEN IF THEIR PROPERTIES CHANGE.

VALUE OBJECTS: OBJECTS WITHOUT IDENTITY, DEFINED ONLY BY THEIR ATTRIBUTES
THEY ARE IMMUTABLE.

AGGREGATES: CLUSTERS OF ENTITIES AND VALUE OBJECTS AS A SINGLE UNIT, WITH AN "AGGREGATE ROOT" GOVERNING THEIR INTERNAL COHERENCE.

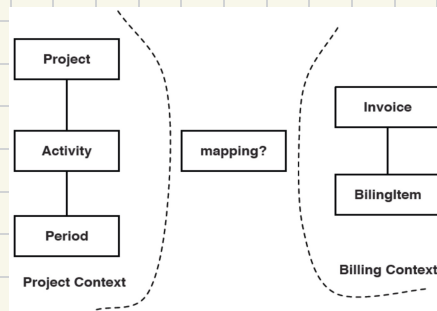
DOMAIN SERVICES: THEY CONTAIN BUSINESS LOGIC THAT DOESN'T NATURALLY BELONG TO AN ENTITY OR VALUE OBJECT. THEY ARE STATELESS.

FACTORIES: OBJECTS RESPONSIBLE FOR CREATING OTHER COMPLEX OBJECTS

REPOSITORIES: THEY MANAGE THE PERSISTENCE AND RECOVERY OF DOMAIN OBJECTS, ACTING AS INTERMEDIARIES BETWEEN THE DOMAIN AND THE DATA LAYER.

TOOLS AND TECHNIQUES

CONTEXT MAPPING: MAP CONTACT POINTS AND TRANSLATIONS BETWEEN DIFFERENT BOUNDED CONTEXTS.



COMMAND QUERY RESPONSABILITY SEGREGATION (CQRS): SEPARATE COMMANDS (STATE CHANGES) FROM QUERIES (STATE READING), IMPROVING SYSTEM FLEXIBILITY AND PERFORMANCE.

