

# **Tema 1 Introducció al procés de desenvolupament del software**

**Maria Salamó Llorente**

**Disseny de Software**

Enginyeria Informàtica

Facultat de Matemàtiques, Universitat de Barcelona

## 1.1 Introducció

## 1.2 Procés de desenvolupament del software

## 1.3 Desenvolupament iteratiu

## 1.4 Fases, iteracions, disciplines i artefactes

## 1.5 Artefactes en el desenvolupament d'una aplicació

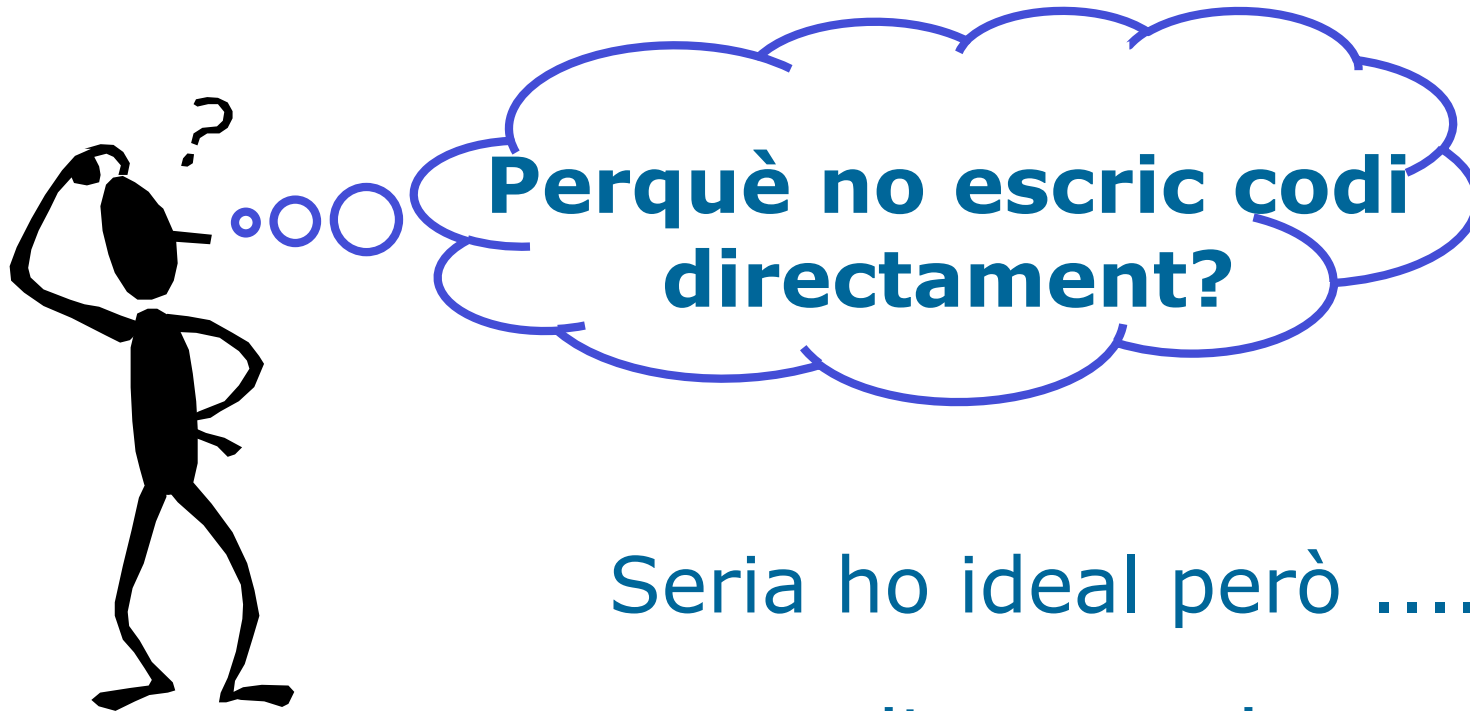
- Casos d'ús
- Model de domini
- Model de disseny
  - Diagrames de interacció
  - Diagrames de classes

# 1.1 Introducció

# Exemple de desenvolupament



# Desenvolupament del SW



Seria ho ideal però ....  
.... necessitem escriure models



Perquè la majoria  
d'empreses no  
practiquen el modelat?

## S'obtenen beneficis amb el modelat?

- Té un cost en **formació** i **temps**
- Millora la **productivitat**
- Millora la **qualitat** del software

# Utilitat del modelat

- Hi ha estructures que no són visibles en els programes.
- Ajuda a raonar sobre el **com s'implementa**
- Facilita la **comunicació** entre l'equip ja que tenen un llenguatge comú
- Es disposa de **documentació** que perdura més enllà del projecte.
- **Generació de codi** a partir de models
  - Ha sorgit un **nou paradigma** de desenvolupament software a partir de models (p.e. MDA de OMG)

# Què és un model?

*"Un model és una simplificació de la realitat"*

*"Un model és resultat d'un procés d'abstracció i ajuda a comprendre i a raonar sobre una realitat"*

## Què és un model software?

*"Un model és una descripció d'un aspecte del sistema, expressada en un llenguatge ben definit"*



# UML i el modelat

*UML és un llenguatge per **visualitzar**, **especificar**, **construir** i **documentar** els artefactes (models) d'un sistema que involucra una gran quantitat de software, des d'una perspectiva **orientada a objectes**.*

- **UML és una notació**, no és un procés
- S'han definit molts processos per a UML
  - *Rational* ha ideat *RUP*, el "procés unificat"
- Usable per sistemes que no siguin software

# Models en UML

- Modelat de **Casos d'ús**
  - Diagrames de Casos d'ús
- Modelat **Estructural**
  - Diagrames de Classes
- Modelat de **Comportament**
  - Diagrames d'Interacció
  - Diagrames d'Estats
- Modelat de flux d'**Activitats**
  - Diagrames d'activitats
- Modelat d'**Implementació**
  - Diagrames de Components
- Modelat de **Desplegament**
  - Diagrames de Desplegament

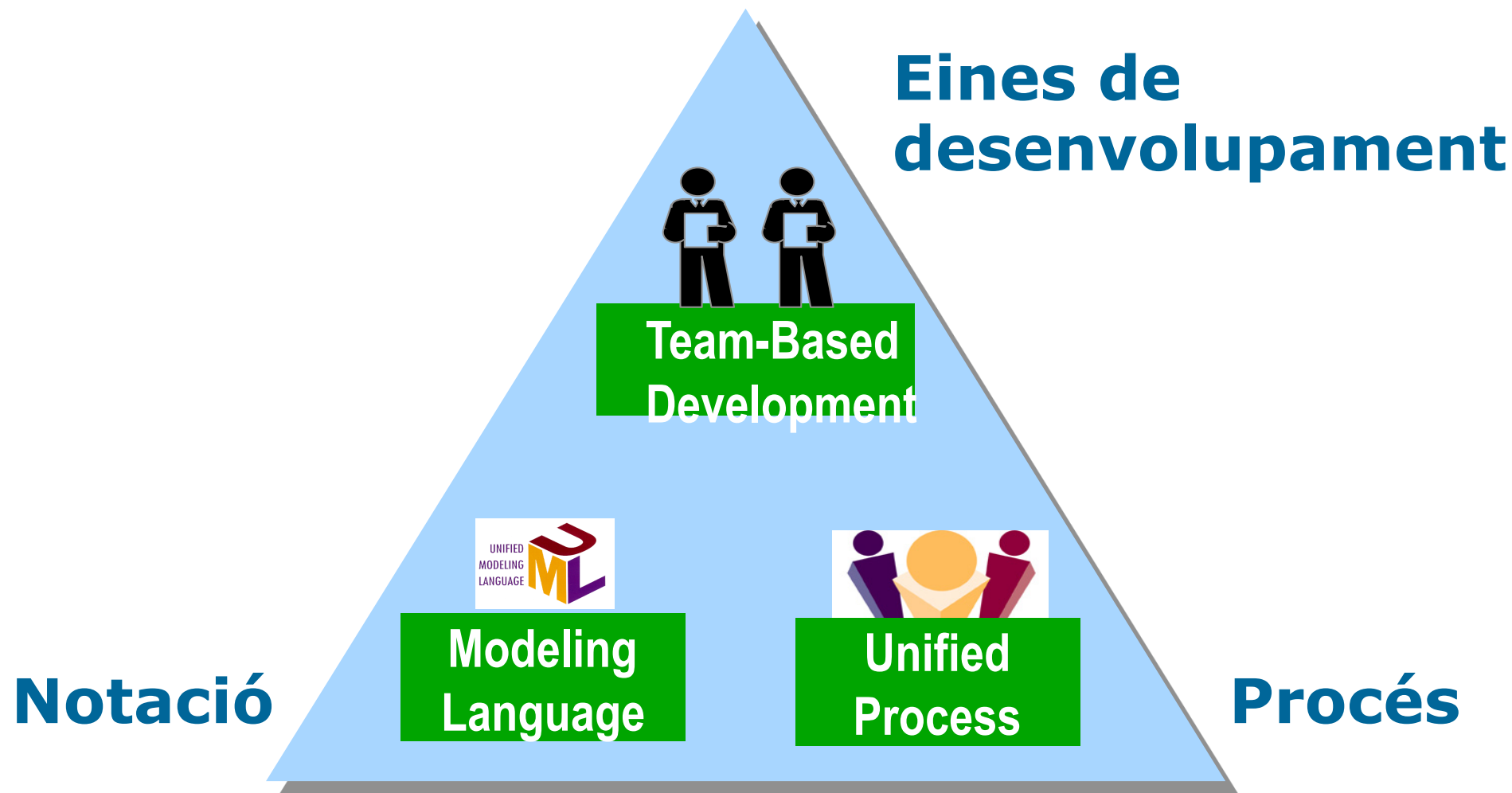


Els diagrames  
no són models

## 1.2 Procés de desenvolupament del SW

# Com desenvolupar el SW?

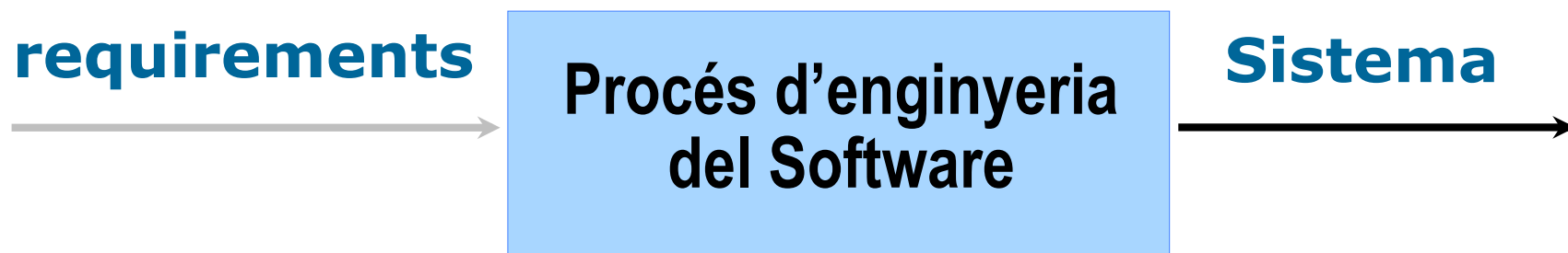
**Però, amb UML no és suficient**



# Què és un procés?

Defineix **qui** està fent **què**, **quan** s'ha de fer i **com** arribar a un cert objectiu

Un procés de desenvolupament del software defineix com **construir**, **desenvolupar** i **mantenir** software



# Beneficis del procés

*"Un procés ben definit es necessari per a desenvolupar sistemes software de manera **repetible i predecible**"*

*"Permet un negoci sostenible i que pot millorar a cada nou projecte, incrementant **l'eficiència** i la **productivitat** de l'organització"*

**G. Booch**

# El procés unificat

No hi ha un únic procés de desenvolupament!

**El procés unificat és:**

- Procés orientat a objectes
- Dirigit per casos d'ús
- Centrat en l'arquitectura
- Amb un cicle de vida *iteratiu* i *incremental*
- Configurable: Té flexibilitat i permet l'extensió (extensibilitat)
- Impulsa un control de qualitat i una gestió del risc d'objectius i continus

# El procés unificat

- El **procés unificat** és molt popular per a la construcció de sistemes orientats a objecte
- El **procés unificat** inclou un conjunt de “*bons costums*” per desenvolupar el software

## The Unified Modeling Language



## The Unified Process

- **OMG standard**
- **Convergència en el futur**
- **Convergència a través de frameworks**

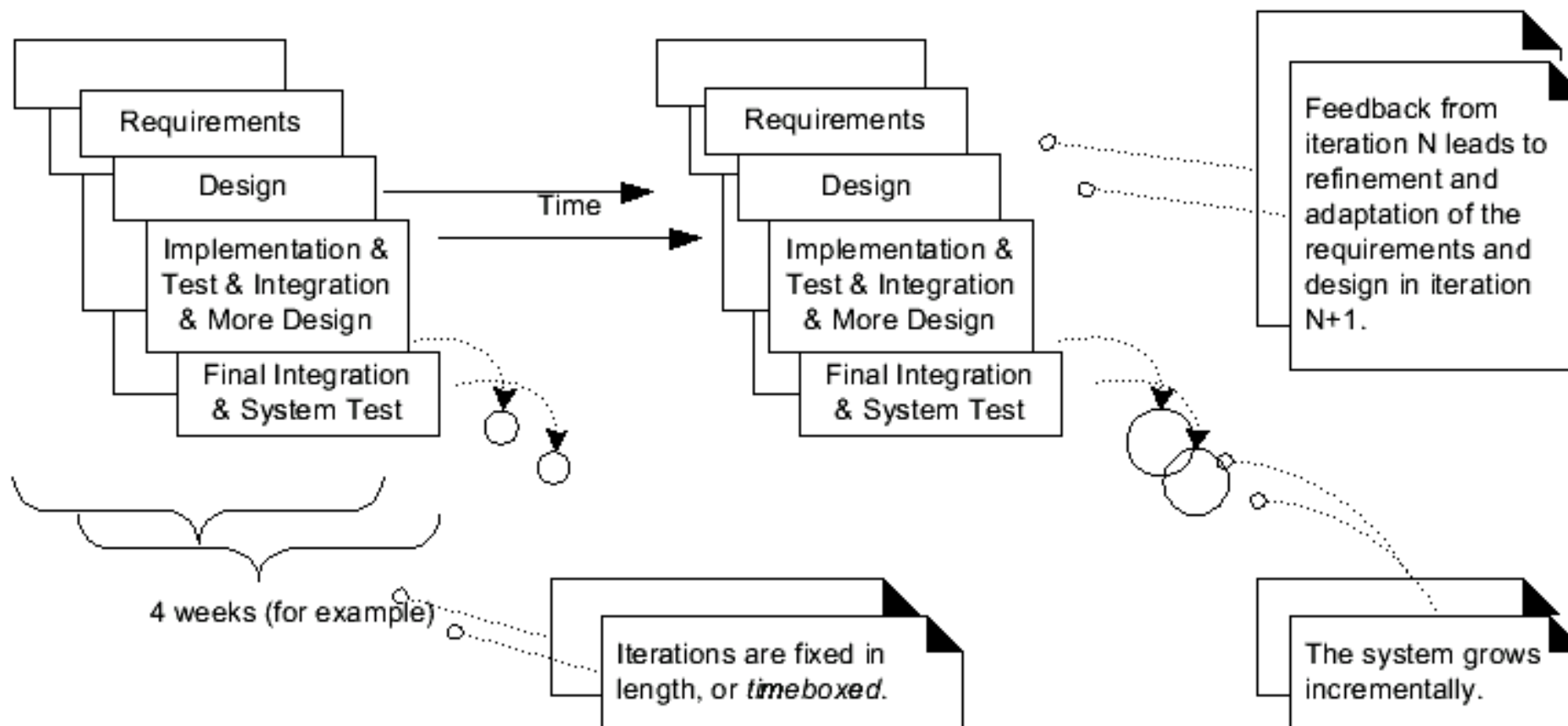


## 1.3 Desenvolupament iteratiu

# Definició desenvolupament iteratiu

- El projecte s'organitza en una sèrie de mini projectes d'un temps limitat i curt (unes 4 setmanes) que s'anomenen **iteracions**.
  - Cada iteració inclou el seu propi anàlisi de requisits, disseny, implementació i testeig
- El sistema creix de forma **incremental**, iteració a iteració, adaptant els seus objectius a canvis en les **necessitats** que es detecten durant la iteració
- No es comença el projecte directament programant però tampoc s'espera a finalitzar un llarg procés d'anàlisi i disseny abans de començar a programar

# Evolució desenvolupament iteratiu



# Iteracions

- El **resultat de cada iteració** és un sistema executable però incomplet, no està llest per passar a producció. Possiblement no ho estigui fins passades 10 o 15 iteracions
- El **resultat** de cada iteració **NO és un prototipus** per llençar, és un software de qualitat amb un subconjunt de la funcionalitat desitjada
- **Cada iteració aborda nous requisits** i amplia el sistema incrementalment. Però, una iteració podria ocasionalment, tornar sobre el software que ja existeix i millorar-lo

# Conclusió

- ① El desenvolupament iteratiu entén **el canvi i l'adaptació** com **inevitables** i essencials
- El desenvolupament d'un projecte consisteix per tant en:
  - Una sèrie de cicles “**construir, rebre feedback – adaptar**”. En les primeres iteracions, la desviació del “veritable objectiu” serà més gran que a les últimes

# Beneficis

- **Ràpida mitigació** dels principals riscos als que s'enfronta el projecte
- Possibilitat de visualitzar el progrés
- **Ràpida recepció de feedback** dels usuaris i adaptació al mateix
- **Complexitat limitada** a cada iteració

## 1.4 Fases, iteracions, disciplines i artefactes

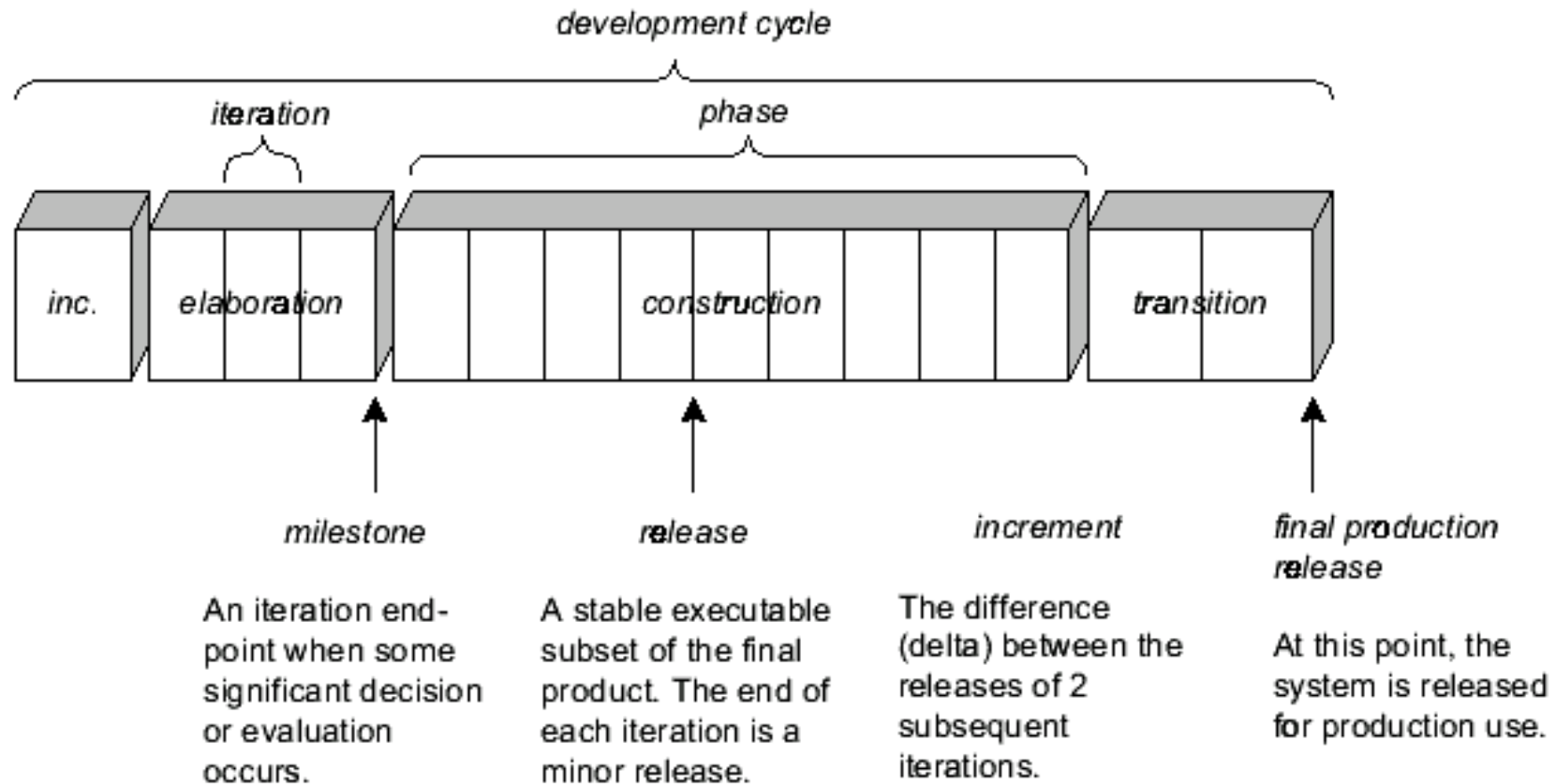
# Fases del cicle de vida

- **Començament o Inici:** Visió aproximada del projecte, anàlisi de negoci, identificació de riscos, planificació inicial, estimacions imprecises
- **Elaboració:** Visió refinada, implementació iterativa de l'arquitectura fonamental, resolució de riscos principals, identificació de la majoria de requisits, estimacions més realistes
- **Construcció:** Implementació iterativa de la resta d'elements de menor risc i més senzills. Desenvolupar el producte i preparació per a l'entrega
- **Transició:** Beta testing, entrega.



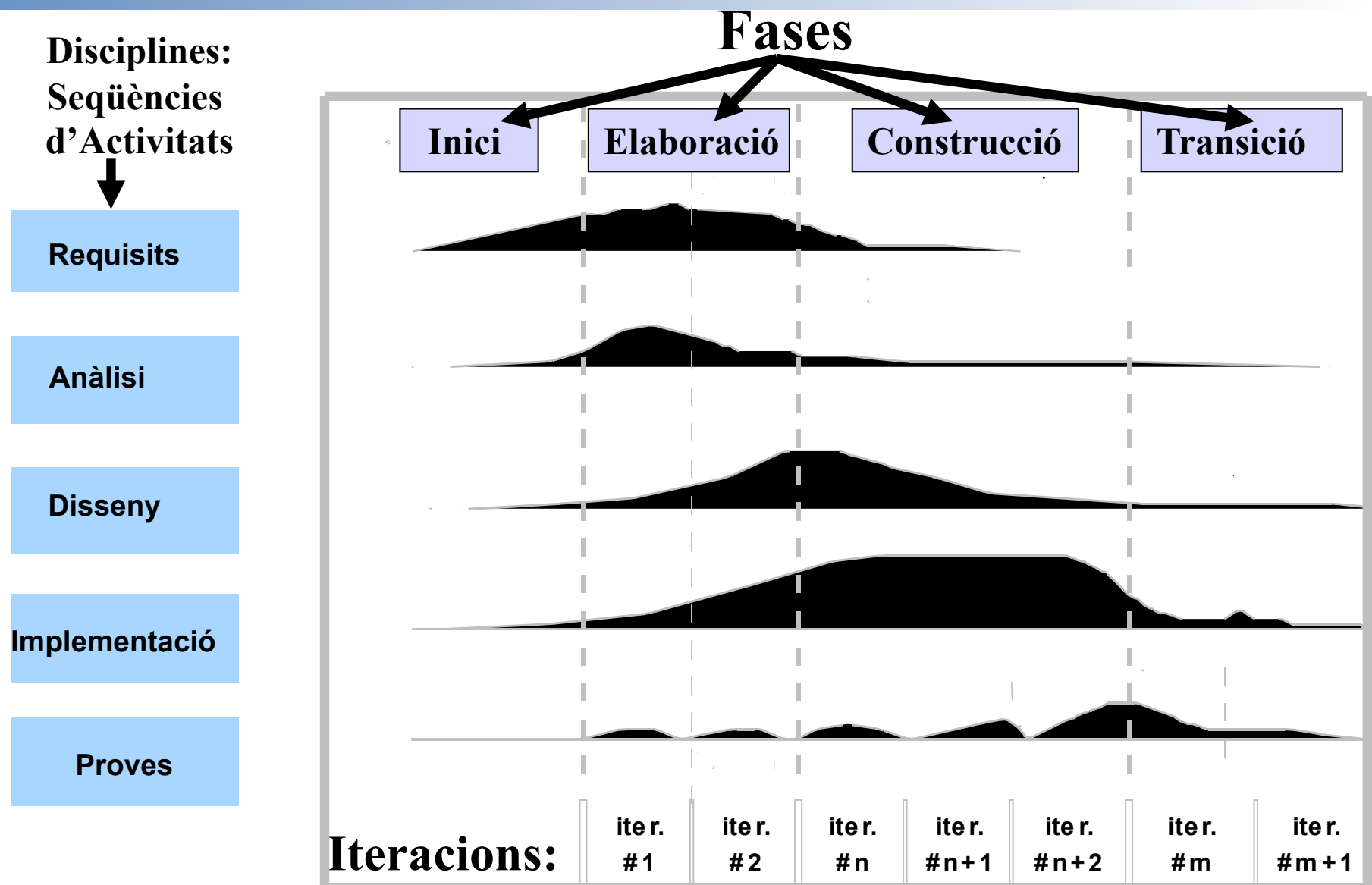


# Fases i iteracions



Una **iteració** és una *seqüència d'activitats* amb un pla establert i un criteri d'avaluació, resultant en una versió estable i executable

# El cicle de vida del procés unificat



# Artefactes i disciplines

- Un **artefacte** és qualsevol producte del treball: codi, esquema de bases de dades, diagrames, models, un pla de projecte, etc.
- Una **disciplina** és un conjunt d'activitats (i artefactes relacionats) en una de les àrees en les que es treballa, com per exemple, les activitats dins de l'àrea d'anàlisi de requisits

# Disciplines

- Aquest curs ens centrarem en **3 disciplines**:
  - **Modelat de Negoci**: Pot anar des d'un model dels objectes de negoci fins a un model dinàmic dels processos d'una empresa
  - **Modelat dels Requeriments**: Comprèn l'anàlisi de requisits per una aplicació. Fonamentalment escriure casos d'ús i identificar els requisits no funcionals
  - **Modelat del Disseny**: Comprèn tots els aspectes de disseny, incloent-hi l'arquitectura general, objectes, bases de dades, xarxes, patrons, etc.

# Fase d'inici

- Quina és la visió i **l'objectiu de negoci** per al projecte?
- És el projecte viable?
- S'ha de construir o comprar el SW?
- Quina és l'estimació de cost?
  - De 30.000 a 60.000€
  - De 300.000 a 400.000€
- Tenint en compte totes les preguntes,
  - **Posem en marxa el projecte?**

# Artefactes que s'inicien en la fase d'inici

- **Visió** i pla de negoci
- **Model de casos d'ús**: requisits funcionals
- **Especificacions addicionals**: requisits no funcionals
- **Llista de riscos** i pla de gestió de riscos
- **Pla d'iteració**: Descriu amb cert detall el que es farà a la propera iteració
- **Pla de fases**: Tracta de preveure l'evolució del projecte més a llarg termini
- **Carpeta de desenvolupament**: Descriu com s'adaptarà el procés unificat al projecte

# Fase d'elaboració

- És la sèrie inicial d'iteracions durant la qual:
  - Es descobreixen i estableixen la major part dels requisits
  - S'eliminen o mitiguen els principals riscos
  - Els principals elements de l'arquitectura s'implementen i es proven

**Construeix l'arquitectura, resol els elements d'alt risc, defineix la majoria dels requisits i estima el programa de treball i els recursos necessaris**

# Recomanacions en la fase d'elaboració

- Fer iteracions de temps fixat i curtes guiades pel risc
- Començar a programar aviat
- Dissenyar, implementar i provar de forma adaptativa el nucli i les parts de major risc de l'arquitectura
- Provar aviat, sovint i de forma realista
- Adaptar-se en funció del feedback dels testers, usuaris i desenvolupadors



# Prioritzant iteracions en la fase d'elaboració

S'han d'ordenar els requisits a cobrir a cada iteració tenint en compte:

- **Risc:** Inclou la complexitat tècnica i altres factors, com la incertesa en l'esforç o la usabilitat
- **Cobriment:** Totes les parts fonamentals del sistema s'han de tocar a les primeres iteracions
- **Criticitat:** Les funcions d'alt valor de negoci que han d'ésser prioritzades

# Artefactes que s'inicien en la fase d'elaboració

- **Model de domini:** És una visualització dels conceptes del domini i les seves relacions
- **Model de disseny:** Descriu el disseny lògic. Inclou diagrames de classe, diagrames d'interacció, diagrames de paquets, etc.
- **Model de dades:** Inclou els esquemes de les BD's i les estratègies per al mapeig objecte-relacional
- **Model de test:** És una descripció del què es provarà i com
- **Model d'implementació:** És la implementació: codi, executable, base de dades, etc.
- **Storyboards, wireframes:** Esquemes de la interfície d'usuari

# Fase de construcció: Desenvolupar el sw

- Es construeix el producte. En aquesta fase:
  - L'arquitectura es completa per construir un sistema ben cimentat
  - La visió evoluciona fins convertir-se en un producte preparat per als usuaris.
  - És la fase on es dediquen la majoria dels recursos
  - L'arquitectura del sistema és estable tot i que es poden fer canvis mínims.
  - El producte s'ajusta a les necessitats d'alguns usuaris com per a enviar-los el software?

# Fase de transició: Proporcionar el sistema als usuaris finals

- El producte es troba en fase *beta*
  - Un grup reduït d'usuaris experimentats prova el producte i informa dels defectes, deficiències i suggereix millores
  - Els desenvolupadors arreglen les deficiències i incorporen alguna de les millores proposades per un major grup d'usuaris
  - Aquesta fase inclou activitats com la venda, formació d'usuaris, oferiment d'ajuda en línia i correcció de defectes descoberts després de la implantació.

# El procés unificat àgil

- És millor un conjunt petit d'activitats i artefactes. El procés ha de ser **simple**.
- Donat que el procés és iteratiu, els requisits i els dissenys no estan complets abans de la implementació. Emergeixen a través d'una sèrie d'iteracions basades en el "**feedback**"
- No existeix un pla detallat per tot el projecte. Existeix un pla d'alt nivell (**pla de fases**) que estima la data final del projecte i altres fites principals. Un pla detallat (**pla d'iteració**) només planeja en detall la propera iteració

# Resum

- El procés unificat de desenvolupament del software és un procés *iteratiu* i *adaptatiu*
- Es **composa de**:
  - **Fases**: Inici, Elaboració, Construcció i Transició
  - **Iteracions**
  - **Disciplines**: Modelat de negoci, requeriments, disseny, etc.
  - **Artefactes**
- El procés unificat tracta de ser un procés àgil

## 1.5 Artefactes

# Artefactes

- Casos d'ús
- Model de domini
- Model de disseny
  - Diagrames de interacció
  - Diagrames de classes

**NOTA!** En els següents apartats es veu una **breu** introducció a cadascun dels punts. Més endavant hi ha un tema que descriu en profunditat cadascun d'ells.



# Casos d'ús

- Els casos d'ús són històries sobre com usar un sistema per satisfer uns objectius
- Descriuen el que fa el sistema des del punt de vista d'un observador extern

Emfatitzen el **què** enlloc del **com**

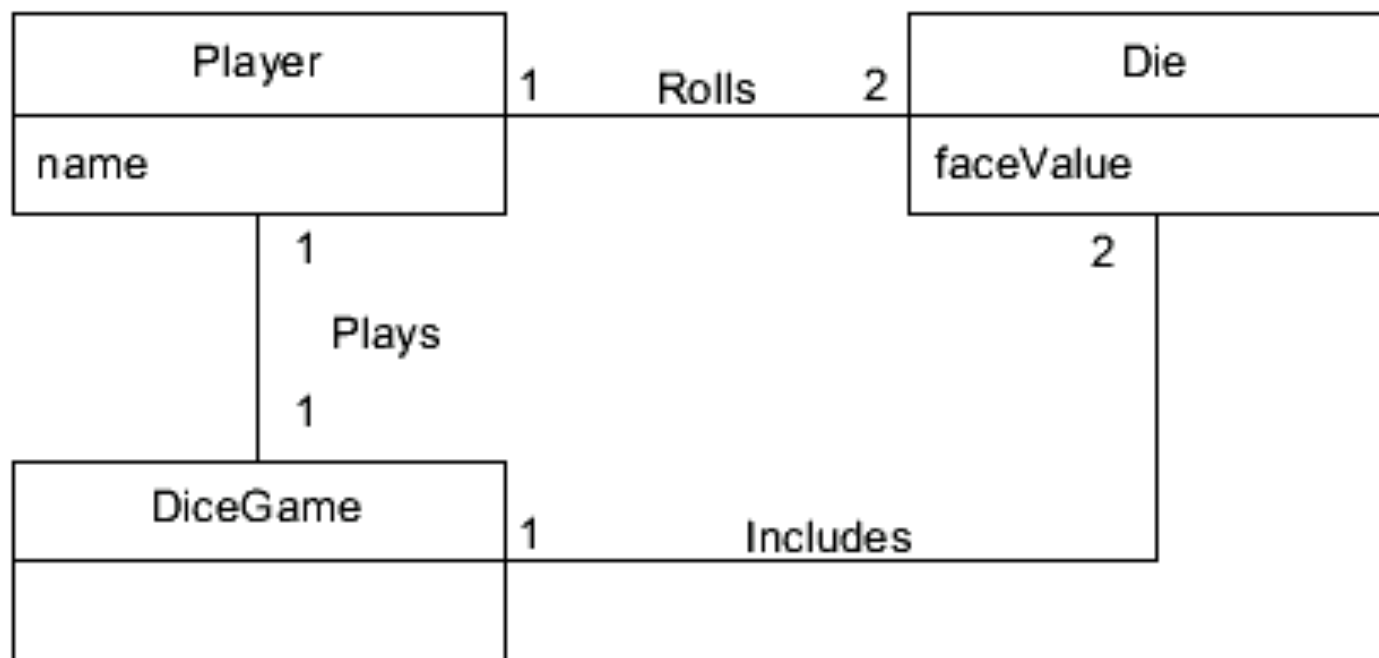
Plantegen **escenaris**, el que passa quan algú interactua amb el sistema. Proporcionen un resum per a un objectiu

**“Jugar a daus:** Un jugador llança 2 daus de sis cares. Si el valor de la suma dels punts representats a la cara superior d'ambdós és 7, el jugador guanya; altrament, perd”

# Model de domini

- Crear **una descripció del domini** des de la perspectiva dels conceptes existents en el mateix
- Comprendre fonamentalment la identificació de:
  - Conceptes
  - Associacions
  - Atributs

# Model de domini

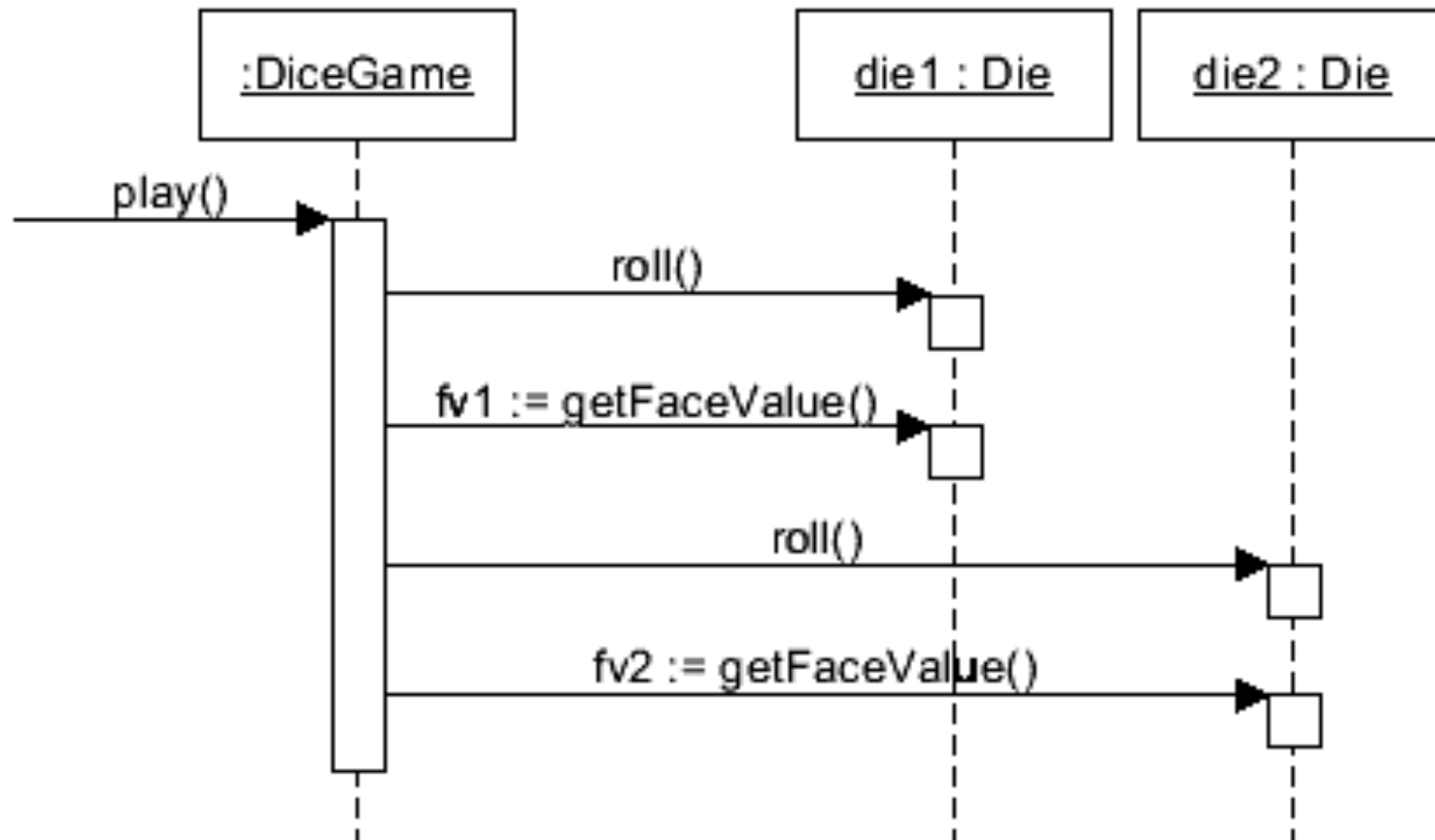


# Model de disseny: diagrames d'interacció

## Visió dinàmica

- Mostren el **flux de missatges** entre objectes i la invocació de mètodes (visió dinàmica)
- Aquest flux no sempre correspon amb el del món real

# Diagrames d'interacció



## Visió estàtica

- Mostren una **visió estàtica** de les definicions i les classes, il·lustrant els seus atributs, els seus mètodes i les seves associacions
- En contrast amb el model de domini, aquest **no mostra conceptes** del món real sinó **classes software**

# Diagrama de classes de disseny

