# The S of SOLID

## The Single Responsibility Principle

Maximilian Meffert

## The Single Responsibility Principle

- Probably around since the 1970’s
  - David L. Parnas, *On the Criteria To Be Used in Decomposing Systems into Modules*. Commun. ACM 15(12): 1053-1058 (1972)
- What does it say?
  - A module / class should only have one responsibility
  - A module / class should only have on reason to change (Robert C. Martin)
- What does it mean?
  - Separate things which are likely to change because of different reasons
  - Group things together which are likely to change for the same reason

## The Single Responsibility Principle

- Why is it a “good” thing to adhere to?
  - Change of requirements is immanent through the life cycle of most software
  - Decreases change impact, i.e. number of modules to alter
  - Decreases risk of regression because of human error, i.e. developer faults
    - *… uhm, protects software against developers?*
  - Increases a software designs capability for adaption to change
    - *… uhm, “agile”?*

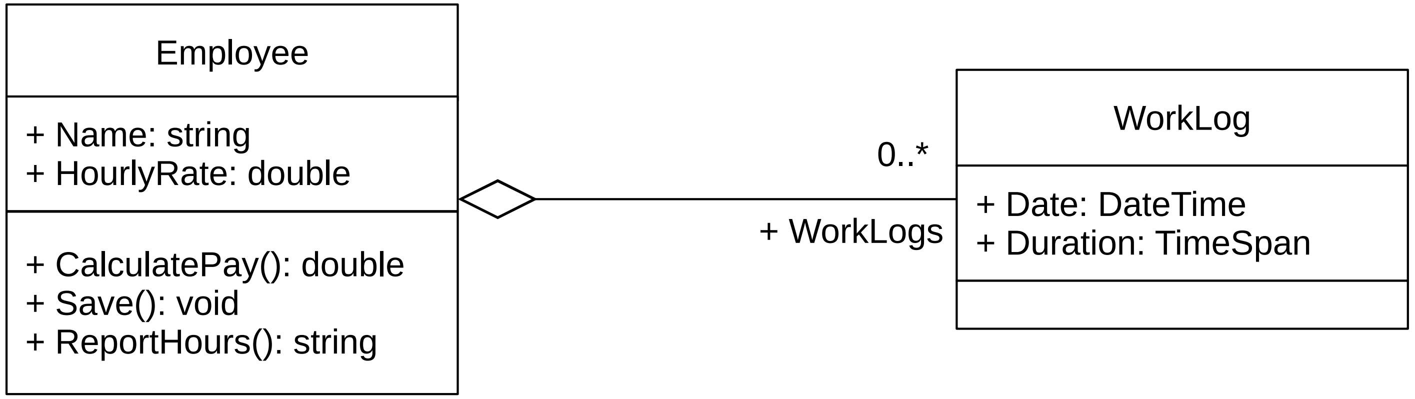
## The Single Responsibility Principle

- What is a “reason to change”?
  - Changes to software are necessary because…
    a) … requirements have literally changed
    b) … of bugs, where requirements have not been met
  - Requirements originate from stake holders
  - Stake holders are who software must ***respond*** to
    - Hence: ***Responsibility*** Principle

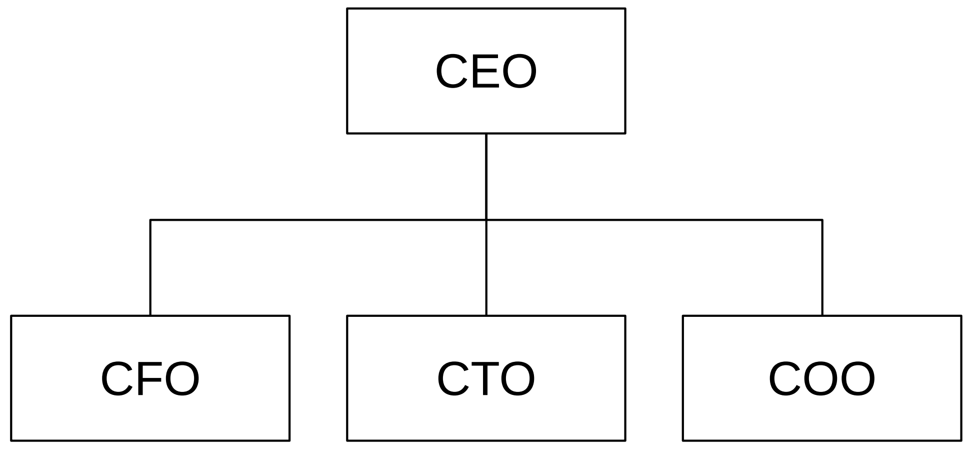
## Example

The Acme Corp. Employee Management System

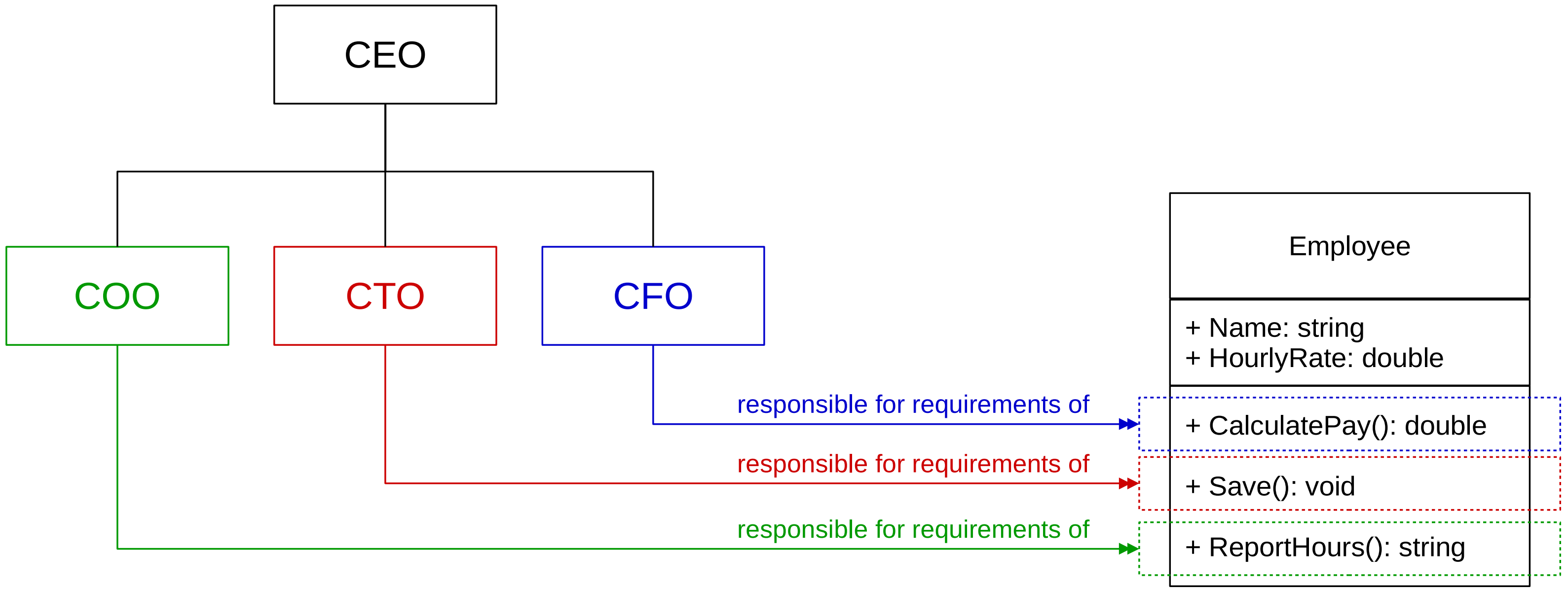
## Acme Corp. Employee Management System

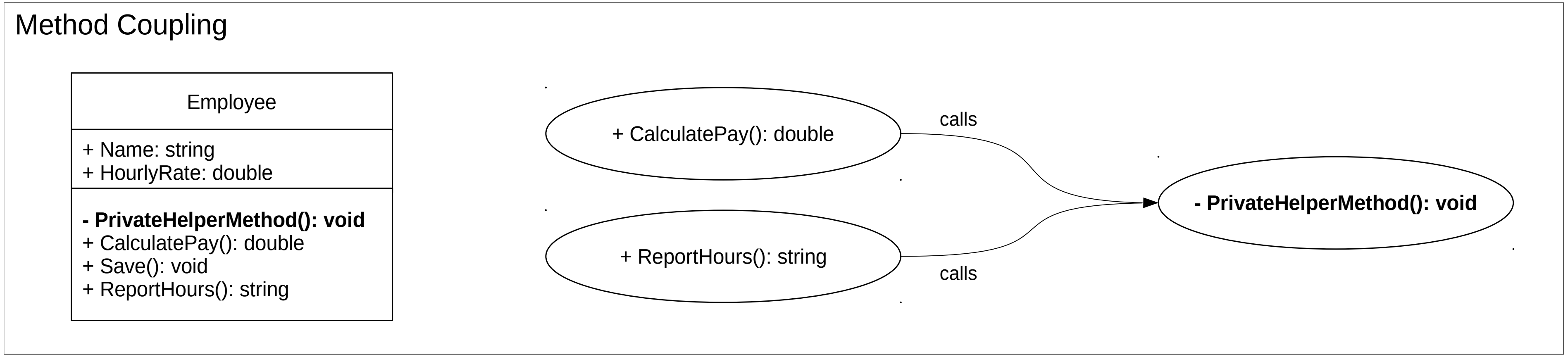


Acme Corp. Organization Chart

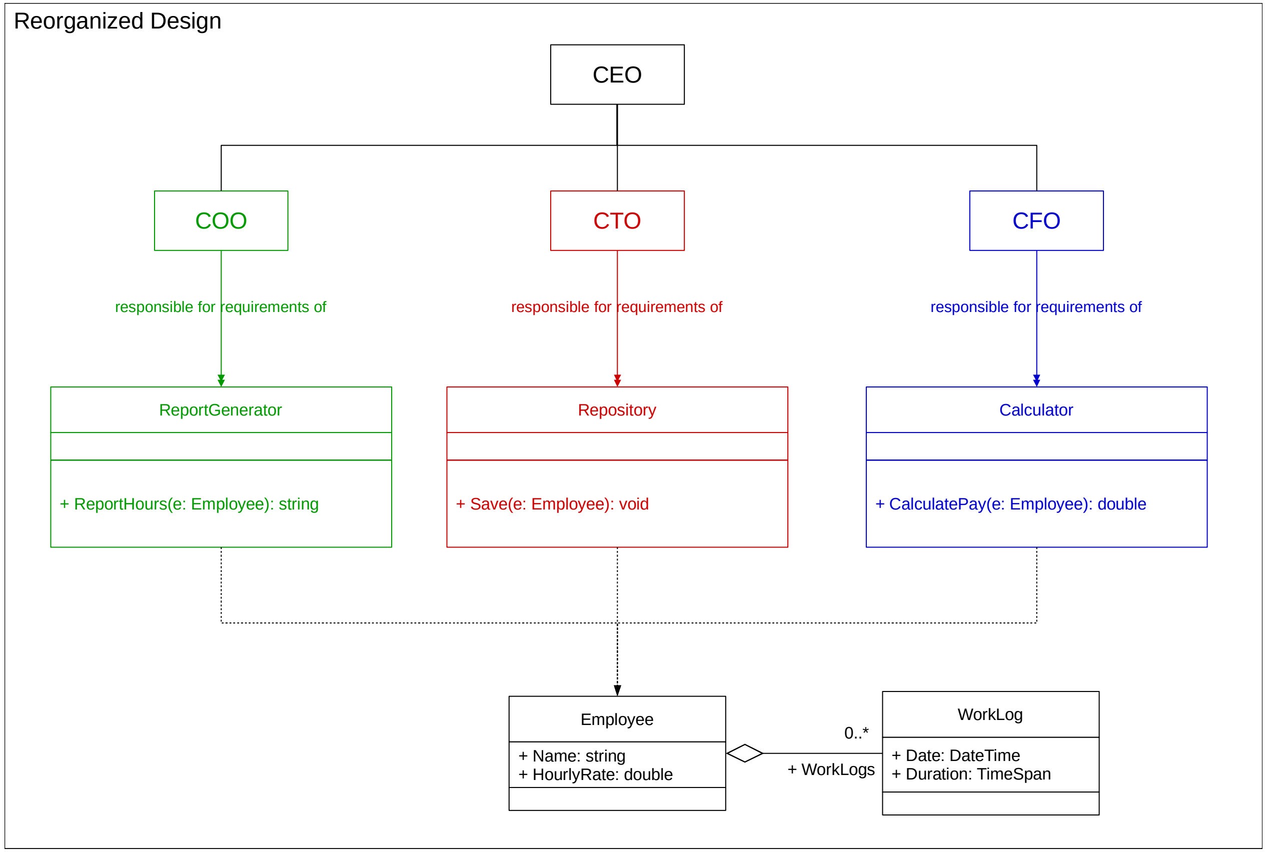


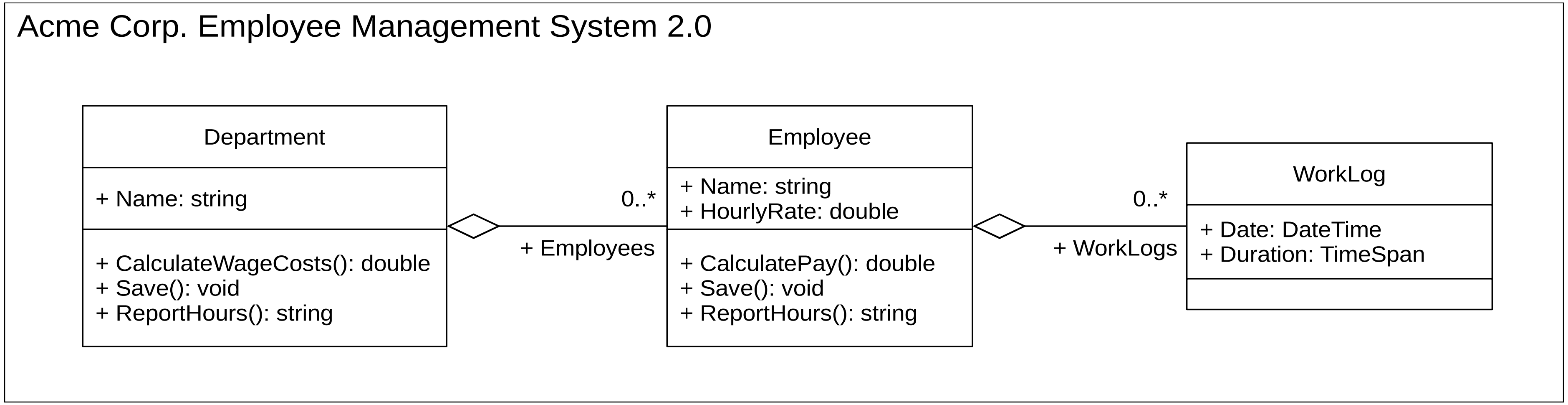
## Responsibilities / Reasons for Change

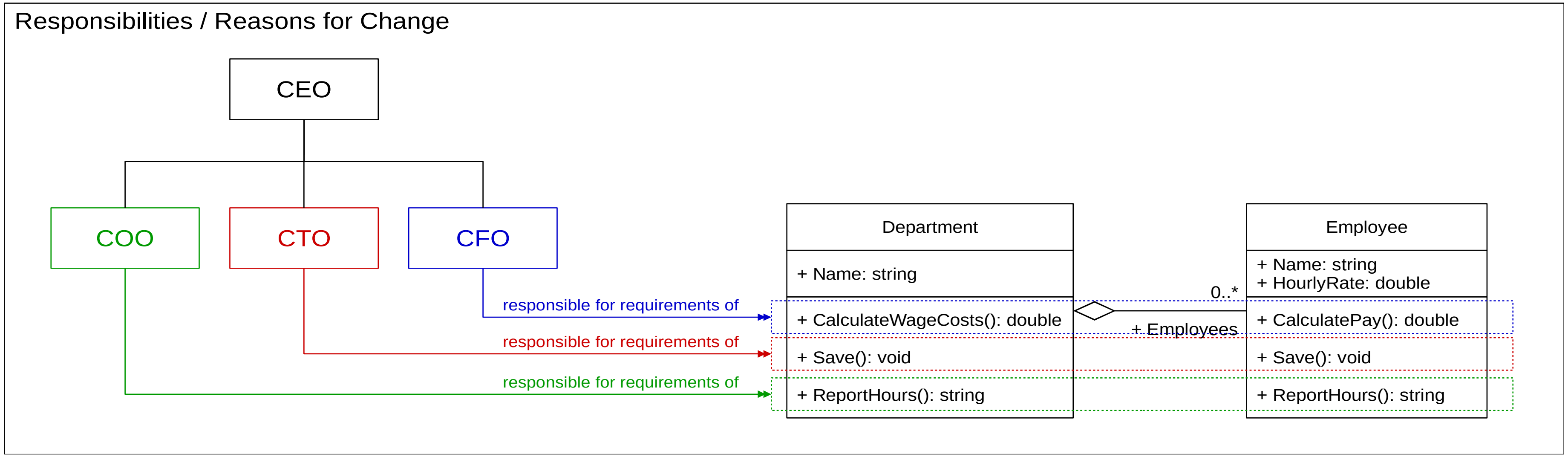


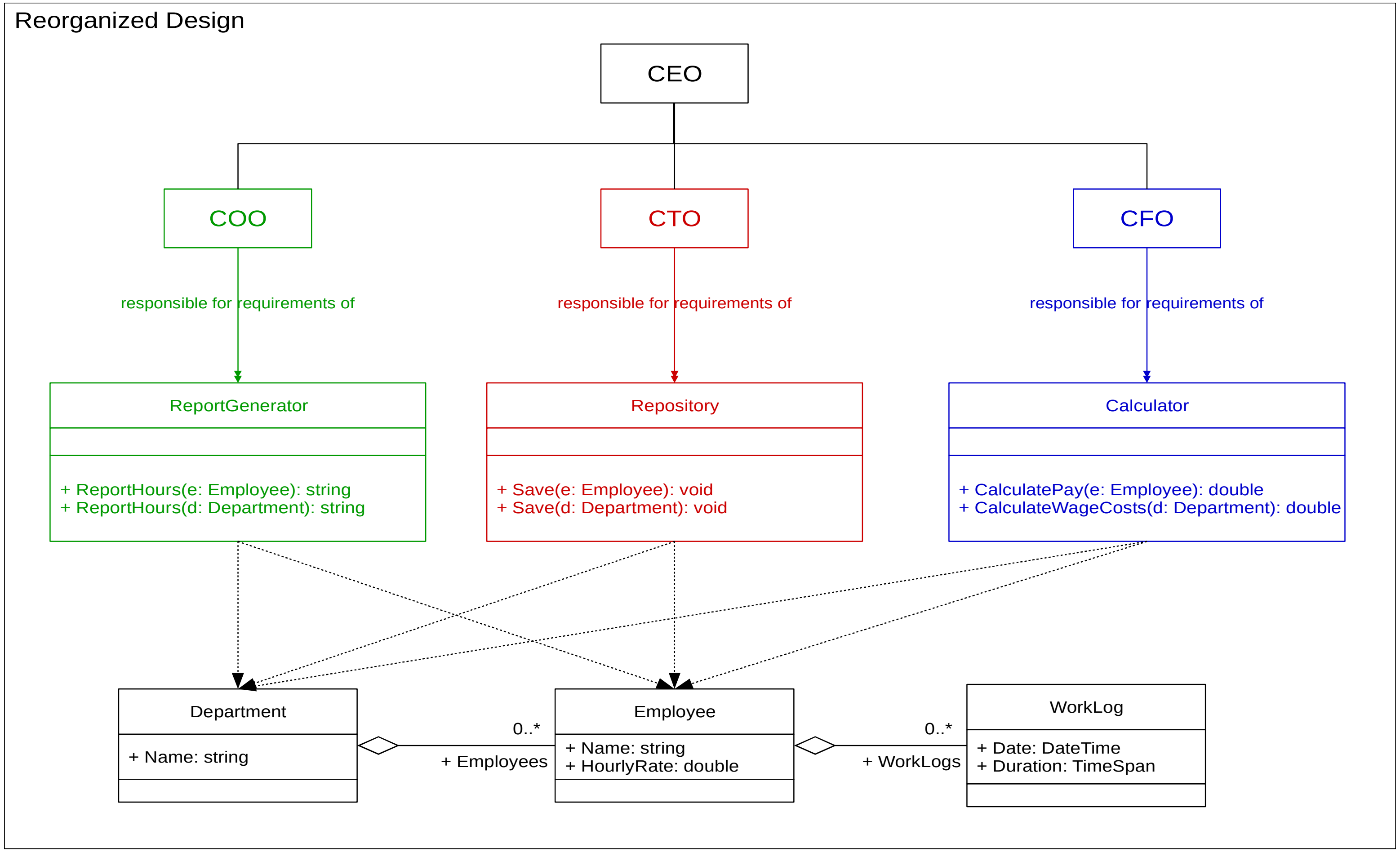


Aggregation of responsibility may lead to hidden coupling which facilitates human error / developer faults


Reorganized Design
CEO
COO
CTO
CFO
responsible for requirements of
responsible for requirements of
responsible for requirements of
ReportGenerator
+ ReportHours(e: Employee): string
Repository
+ Save(e: Employee): void
Calculator
+ CalculatePay(e: Employee): double
Employee
+ Name: string
+ HourlyRate: double
0..*
+ WorkLogs
WorkLog
+ Date: DateTime
+ Duration: TimeSpan


Acme Corp. Employee Management System 2.0
Department
+ Name: string
+ CalculateWageCosts(): double
+ Save(): void
+ ReportHours(): string
0..*
+ Employees
Employee
+ Name: string
+ HourlyRate: double
+ CalculatePay(): double
+ Save(): void
+ ReportHours(): string
0..*
+ WorkLogs
WorkLog
+ Date: DateTime
+ Duration: TimeSpan


Responsibilities / Reasons for Change
CEO
COO
CTO
CFO
responsible for requirements of
responsible for requirements of
responsible for requirements of
Department
+ Name: string
+ CalculateWageCosts(): double
+ Save(): void
+ ReportHours(): string
0..*
+ Employees
Employee
+ Name: string
+ HourlyRate: double
+ CalculatePay(): double
+ Save(): void
+ ReportHours(): string



Reorganized Design
CEO
COO
CTO
CFO
responsible for requirements of
responsible for requirements of
responsible for requirements of
ReportGenerator
+ ReportHours(e: Employee): string
+ ReportHours(d: Department): string
Repository
+ Save(e: Employee): void
+ Save(d: Department): void
Calculator
+ CalculatePay(e: Employee): double
+ CalculateWageCosts(d: Department): double
Department
+ Name: string
0..*
+ Employees
Employee
+ Name: string
+ HourlyRate: double
0..*
+ WorkLogs
WorkLog
+ Date: DateTime
+ Duration: TimeSpan

Thanks!