

# Papers Fundacionales

## 7 Mitos acerca de los Métodos Formales

Foguelman - Modrow - Tilli

DC - UBA

1 de julio de 2013

# Introducción

- Artículo aparece en IEEE Software, September 1990
- Anthony Hall
  - ingeniero de software y sistemas
  - especializado en métodos formales
  - químico
- Trabajaba en proyecto CASE (herramienta para SSADM) al momento del artículo

# Mito 1: Pueden garantizar que el software es perfecto

- Las demostraciones son falibles

# Mito 1: Pueden garantizar que el software es perfecto

- Las demostraciones son falibles
- Límite en las demostraciones (mundo real vs modelo)

# Mito 1: Pueden garantizar que el software es perfecto

- Las demostraciones son falibles
- Límite en las demostraciones (mundo real vs modelo)
- Error en la especificación

# Mito 1: Pueden garantizar que el software es perfecto

- Las demostraciones son falibles
- Límite en las demostraciones (mundo real vs modelo)
- Error en la especificación
- + Demostración de correctitud: propiedades globales, y relación entre programa y especificación

# Mito 1: Pueden garantizar que el software es perfecto

- Las demostraciones son falibles
- Límite en las demostraciones (mundo real vs modelo)
- Error en la especificación
- + Demostración de correctitud: propiedades globales, y relación entre programa y especificación
- + Exponer errores

# Mito 1: Pueden garantizar que el software es perfecto

- Las demostraciones son falibles
- Límite en las demostraciones (mundo real vs modelo)
- Error en la especificación
- + Demostración de correctitud: propiedades globales, y relación entre programa y especificación
- + Exponer errores

**Hecho:** Los métodos formales son muy útiles para exponer errores de manera temprana y pueden eliminar casi completamente algunas clases de errores



## Mito 2: Son útiles sólo para probar correctitud

- + Ayuda a la clarificación de requerimientos

## Mito 2: Son útiles sólo para probar correctitud

- + Ayuda a la clarificación de requerimientos
- + Construcción de la especificación y detección temprana de errores

## Mito 2: Son útiles sólo para probar correctitud

- + Ayuda a la clarificación de requerimientos
- + Construcción de la especificación y detección temprana de errores
- + Demostración de propiedades de la especificación

## Mito 2: Son útiles sólo para probar correctitud

- + Ayuda a la clarificación de requerimientos
- + Construcción de la especificación y detección temprana de errores
- + Demostración de propiedades de la especificación
- + Implementación semiautomática e iterativa

## Mito 2: Son útiles sólo para probar correctitud

- + Ayuda a la clarificación de requerimientos
- + Construcción de la especificación y detección temprana de errores
- + Demostración de propiedades de la especificación
- + Implementación semiautomática e iterativa
  - Prueba de correctitud dificultosa a pesar del uso de metodos formales

**Hecho:** Trabajan generalmente haciendo que pienses mucho sobre el sistema que pretendes construir

## Mito 3: Sólo se benefician los sistemas altamente críticos

- + Se benefician muchos sistemas: críticos, replicados, embebidos en hardware, de alta calidad

## Mito 3: Sólo se benefician los sistemas altamente críticos

- + Se benefician muchos sistemas: críticos, replicados, embebidos en hardware, de alta calidad
- + Objetividad, mantenimiento, facilidad de construcción, visibilidad

## Mito 3: Sólo se benefician los sistemas altamente críticos

- + Se benefician muchos sistemas: críticos, replicados, embebidos en hardware, de alta calidad
- + Objetividad, mantenimiento, facilidad de construcción, visibilidad
- + Monitoreo del desarrollo



## Mito 3: Sólo se benefician los sistemas altamente críticos

- + Se benefician muchos sistemas: críticos, replicados, embebidos en hardware, de alta calidad
- + Objetividad, mantenimiento, facilidad de construcción, visibilidad
- + Monitoreo del desarrollo
- + Distinta granularidad en la formalidad

## Mito 3: Sólo se benefician los sistemas altamente críticos

- + Se benefician muchos sistemas: críticos, replicados, embebidos en hardware, de alta calidad
- + Objetividad, mantenimiento, facilidad de construcción, visibilidad
- + Monitoreo del desarrollo
- + Distinta granularidad en la formalidad

**Hecho:** Son útiles para casi cualquier tipo de aplicación

## Mito 4: Involucran matemática compleja

- Dificultad en modelar el mundo real

## Mito 4: Involucran matemática compleja

- Dificultad en modelar el mundo real
- Muy abstracto o muy específico

## Mito 4: Involucran matemática compleja

- Dificultad en modelar el mundo real
- Muy abstracto o muy específico
- Se necesita un experto, algunas veces un consultor experto

## Mito 4: Involucran matemática compleja

- Dificultad en modelar el mundo real
- Muy abstracto o muy específico
- Se necesita un experto, algunas veces un consultor experto
- + Mayormente sólo se necesita lógica y teoría de conjuntos

## Mito 4: Involucran matemática compleja

- Dificultad en modelar el mundo real
- Muy abstracto o muy específico
- Se necesita un experto, algunas veces un consultor experto
- + Mayormente sólo se necesita lógica y teoría de conjuntos
- + En general especificación más corta y sencilla que implementación

## Mito 4: Involucran matemática compleja

- Dificultad en modelar el mundo real
- Muy abstracto o muy específico
- Se necesita un experto, algunas veces un consultor experto
- + Mayormente sólo se necesita lógica y teoría de conjuntos
- + En general especificación más corta y sencilla que implementación

**Hecho:** Se basan en especificaciones matemáticas que son más fáciles de entender que un programa



## Mito 5: Incrementan los costos de desarrollo

- Difícil de medir avance durante especificación (aunque hay casos de éxito)

## Mito 5: Incrementan los costos de desarrollo

- Difícil de medir avance durante especificación (aunque hay casos de éxito)
- Las especificaciones nunca son perfectas

## Mito 5: Incrementan los costos de desarrollo

- Difícil de medir avance durante especificación (aunque hay casos de éxito)
- Las especificaciones nunca son perfectas
- ? Cambio en el modelo de ciclo de vida

## Mito 5: Incrementan los costos de desarrollo

- Difícil de medir avance durante especificación (aunque hay casos de éxito)
- Las especificaciones nunca son perfectas
- ? Cambio en el modelo de ciclo de vida
- + Evidencia de mejores medidores de productividad

**Hecho:** Pueden decrementar los costos de desarrollo

## Mito 6: Son incomprensibles para los clientes

+ Inclusión de traducción al lenguaje natural

## Mito 6: Son incomprensibles para los clientes

- + Inclusión de traducción al lenguaje natural
- + Demostración de cumplimiento de funcionalidad

## Mito 6: Son incomprensibles para los clientes

- + Inclusión de traducción al lenguaje natural
- + Demostración de cumplimiento de funcionalidad
- + Uso de prototipos, o animar la especificación

## Mito 6: Son incomprensibles para los clientes

- + Inclusión de traducción al lenguaje natural
- + Demostración de cumplimiento de funcionalidad
- + Uso de prototipos, o animar la especificación

**Hecho:** Pueden ayudar a los clientes a entender qué están comprando



## Mito 7: Nadie los usa para proyectos reales

- + Evidencia de uso en varios sistemas y disciplinas:  
procesamiento de transacciones, hardware, compiladores,  
herramientas de desarrollo de software, control de  
reactores

## Mito 7: Nadie los usa para proyectos reales

- + Evidencia de uso en varios sistemas y disciplinas:  
procesamiento de transacciones, hardware, compiladores,  
herramientas de desarrollo de software, control de  
reactores

**Hecho:** Son utilizados con éxito en proyectos útiles de la industria

- Ninguna panacea
- Aplicables a cualquier aplicación, detectan ciertos tipos de errores
- Pueden achicar costos
- Es preciso revisar las metodologías a utilizar

¿Preguntas?